

DESARROLLO DE DISPOSITIVO PORTÁTIL PARA MONITOREAR INFORMACIÓN AMBIENTAL Y DE INTERÉS PARA CICLISTAS

SEBASTIAN GONZALEZ RODRÍGUEZ

MAURICIO MENDOZA MONTOYA

**Trabajo de grado para optar al título de
INGENIERO MECATRÓNICO**

MSc. Andrés Felipe Valle Pérez



**UNIVERSIDAD EIA
INGENIERÍA MECATRÓNICA
ENVIGADO
2018**

AGRADECIMIENTOS

Agradecemos a nuestro director Andrés Valle por su acompañamiento durante todo el desarrollo del trabajo de grado, a nuestro gran director de carrera Georffrey Acevedo por todo su apoyo y a Mateo Ardila por todas las asesorías brindadas durante todo el proyecto.

CONTENIDO

	pág.
INTRODUCCIÓN.....	11
1. PRELIMINARES.....	12
1.1 Planteamiento del problema	12
1.2 Objetivos del proyecto	13
1.2.1 Objetivo General.....	13
1.2.2 Objetivos Específicos	13
1.3 Marco de referencia.....	13
1.3.1 Antecedentes	13
1.3.2 Marco Teórico	14
2. METODOLOGÍA.....	16
2.1 Identificación de los requerimientos del sistema a desarrollar	16
2.2 Diseño de concepto.....	16
2.3 Diseño del detalle	16
2.4 Construcción del prototipo	16
2.5 Pruebas y refinamiento.....	16
3. DESARROLLO DEL PROYECTO	18
3.1 Vigilancia tecnológica	18
3.2 Identificación de requerimientos	20
3.2.1 Comparación con soluciones existentes.....	21
3.2.2 Especificaciones objetivo.....	22
3.3 Diseño y selección de concepto	22
3.3.1 Registro de soluciones	23
3.3.2 Matriz Morfológica	28
3.3.3 Selección de concepto.....	31
3.4 Diseño de detalle.....	33
3.4.1 Página Web.....	33
3.4.2 Diseño Electrónico.....	58
3.4.3 Diseño Mecánico	65
3.5 Pruebas y refinamiento.....	69
4. RESULTADOS	70
4.1 Prototipo funcional.....	70
4.2 Página web.....	72
5. CONCLUSIONES Y CONSIDERACIONES FINALES	78
REFERENCIAS	79

LISTA DE TABLAS

	pág.
Tabla 1. <i>Lista de requerimientos para el dispositivo</i>	20
Tabla 2. <i>Especificaciones Objetivo</i>	22
Tabla 3. <i>Temperatura y Humedad</i>	24
Tabla 4. <i>Ozono:</i>	24
Tabla 5. <i>Partículas</i>	25
Tabla 6. <i>Monóxido de carbono</i>	25
Tabla 7. <i>Dióxido de carbono</i>	26
Tabla 8. <i>Compuestos volátiles orgánicos (COV)</i>	26
Tabla 9. <i>Almacenar en base de datos:</i>	27
Tabla 10. <i>Enviar y recibir información</i>	27
Tabla 11. <i>Procesamiento de datos</i>	28
Tabla 12. <i>Matriz Morfológica</i>	29
Tabla 13. <i>Resultado selección concepto</i>	32
Tabla 14. <i>Información Sensores</i>	33

LISTA DE FIGURAS

	pág.
Figura 1. Caja Negra	23
Figura 2. Caja Transparente	23
Figura 3. Interfaz Inicio de sesión inicial	34
Figura 4. Interfaz sensores inicial	34
Figura 5. Interfaz mapa de calor inicial	35
Figura 6. gcloud.....	35
Figura 7. XAMP	36
Figura 8. SSH.....	36
Figura 9. Inicio de sesión	37
Figura 10. Código <i>Index.html</i> parte 1	38
Figura 11. Código <i>Index.html</i> parte 2	39
Figura 12. Código <i>Index.html</i> parte 3	39
Figura 13. Código <i>loginjs.js</i> parte 1.....	40
Figura 14. Código <i>loginphp.php</i> parte 1	41
Figura 15. Código <i>loginphp.php</i> parte 2	41
Figura 16. Mapa y estado actual del aire	42
Figura 17. Código <i>paginamapa.html</i> parte 1	43
Figura 18. Código <i>paginamapa.html</i> parte 2	44
Figura 19. Código <i>paginamapa.html</i> parte 3	44
Figura 20. Código <i>paginamapa.html</i> parte 4	45
Figura 21. Código <i>paginamapa.html</i> parte 5	46
Figura 22. Código <i>paginamapa.html</i> parte 6	46
Figura 23. Archivo <i>mapdata.geo.json</i>	47
Figura 24. Código <i>modifyjson.php</i> parte1	48
Figura 25. Código <i>modifyjson.php</i> parte2	48
Figura 26. CRON	49
Figura 27. Código <i>gotojs.js</i> parte 1	50
Figura 28. Código <i>gotojs.js</i> parte 2	51
Figura 29. Código <i>datosphp.php</i>	51
Figura 30. Información Sensores	52
Figura 31. Código <i>paginausuario.html</i> parte 1	53
Figura 32. Código <i>paginausuario.html</i> parte 2	54
Figura 33. Código <i>paginausuario.html</i> parte 3	54
Figura 34. Código <i>ingresodatos.php</i> parte 1	55

Figura 35. Código <i>ingresodatos.php</i> parte 2	55
Figura 36. Interfaz de phpMyAdmin	56
Figura 37. Tabla Usuarios.....	56
Figura 38. Tabla Datos	57
Figura 39. Advanced REST Client	57
Figura 40. Conexión Arduino – Atmega	59
Figura 41. Código en Arduino parte 1	60
Figura 42. Código en Arduino parte 2	61
Figura 43. Código en Arduino parte 3	61
Figura 44. Código en Arduino parte 4	62
Figura 45. Código en Arduino parte 5	62
Figura 46. Respuesta por el puerto serial	63
Figura 47. Circuito montado en protoboard.....	63
Figura 48. Circuito impreso.....	64
Figura 49. Diseño estructura dispositivo	65
Figura 50. Proceso corte láser.....	65
Figura 51. Prototipo final.....	66
Figura 52. Modelo 3d de la estructura.....	67
Figura 53. Prototipo acoplado vista 1	67
Figura 54. Prototipo acoplado vista 2.....	68
Figura 55. Prototipo acoplado vista 3.....	68
Figura 56. Arduino Nano V 3.0.....	69
Figura 57. Prototipo acoplado a una bicicleta	70
Figura 58. Prototipo acoplado a una bicicleta 2	71
Figura 59. Interfaz inicial.....	72
Figura 60. Interfaz mapa.....	73
Figura 61. Interfaz Información Sensores	74
Figura 62. Comparación con el Siata por Parque Ecológico Lago Los Patos.....	75
Figura 63. Comparación con el Siata por la vía Palmas.....	76
Figura 64. Comparación con el Siata por el Instituto Salesiano Pedro Justo Berrio	76
Figura 65. Comparación con el Siata en la Universidad Nacional de Colombia	77

LISTA DE ANEXOS

	pág.
Anexo 1. Diseño electrónico	83
Anexo 2. Diseño electrónico zoom 1	84
Anexo 3. Diseño electrónico zoom 2.....	85
Anexo 4. Diseño electrónico zoom 3.....	86
Anexo 5. Diseño electrónico PCB	87

GLOSARIO

PM 2,5: son partículas suspendidas en el aire con un tamaño menor a 2.5 micras que sirven para dar una indicación de la contaminación en la ciudad.

IoT: Internet de las cosas (siglas en inglés), es un sistema en donde se conectan distintos dispositivos con identificadores únicos con la capacidad de transferir datos en una red.

HTML: Lenguaje de marcas de hipertexto (siglas en inglés), Sirve para desarrollar la estructura de una página web.

INTERFAZ: superficie de comunicación entre dos sistemas.

MICROCONTROLADOR: circuito integrado que se puede programar y reprogramar dependiendo de la necesidad, capaz de ejecutar la tarea que tenga guardada en su memoria.

O3: Es un gas generalmente incoloro que en caso de ingerir altas cantidades puede producir tos, inflamación de las vías respiratorias e irritación en los ojos, entre otros síntomas.

CO: Gas perjudicial para la salud, es invisibles e inodoro y si se respira en altas concentraciones puede causar la muerte.

CO2: Gas incoloro y denso que si se respira en altas cantidades puede causar dolores de cabeza, insomnio y nauseas.

API: Interfaz de programación de aplicaciones, es un conjunto de funciones y procedimientos que ofrece una biblioteca para ser utilizados en otros softwares.

GPS: Sistema de posicionamiento global (siglas en inglés), determina la posición globalmente utilizando una red de satélites.

GPRS: Servicio general de paquetes vía radio (siglas en inglés), sirve para la transmisión de datos entre dispositivos a largas distancias de forma inalámbrica.

JAVASCRIPT: Lenguaje de programación que permite desarrollar acciones en una página web.

JSON: Es un formato de texto que se utiliza en javascript para el intercambio de datos.

PETICIONES QUERY: Son instrucciones que se utilizan para hacer consultas en una base de datos.

RESUMEN

Según la organización Mundial de la salud, “se calcula que en el mundo suman 1,3 millones las personas que mueren en un año a causa de la contaminación atmosférica urbana” (Organización Mundial de la Salud, 2018).

La contaminación ambiental es un factor que ha escalado acorde a los modelos económicos que se manejan actualmente, es decir, la producción en masa. Además, el transporte en la ciudad también afecta directamente la calidad de aire; para el año 2015 “en la comuna 14 (El Poblado) estaban rodando 1.252 automotores por cada 1.000 habitantes” (Álvarez, 2015), es decir que había aproximadamente un carro por cada tres personas en ese sector.

Un estudio realizado desde 1990 hasta el 2012 revela que “en Colombia se emitieron 258 millones de toneladas de emisiones de gases directo a la atmosfera como resultado de actividades humanas. Los sectores que más generaron emisiones fueron el forestal con el 36%, el sector agropecuario con el 26%, el sector transporte con un 11%, la industria manufacturera con 11% y el sector de minas y energía con un 10 %” (Correa, 2016).

Ya existen dispositivos encargados de monitorear variables climatológicas referentes a la calidad del aire en el valle de Aburrá. Por lo tanto, se requiere un dispositivo que envíe la información recolectada a la nube para ser procesada y, a diferencia de los otros dispositivos que se tienen actualmente, muestre resultados en un mapa de la ciudad, indicándole al usuario en tiempo real que calidad de aire está respirando en cada zona.

Como es una aplicación dedicada a los ciclistas, el dispositivo también cuenta con sensores de temperatura, humedad y un GPS. Adicionalmente, la red de datos en la nube se está actualizando con cada ruta hecha por cualquier individuo que lleve consigo el dispositivo. Por esta razón, siempre se dispone de datos actualizados de los niveles de contaminación de las diferentes zonas de la ciudad.

En este proyecto se emplea la metodología Ulrich para guiar el desarrollo del proyecto. Este se llevará a cabo en tres (3) etapas principales, correspondientes a los objetivos específicos del proyecto. Estas etapas hacen referencia al desarrollo del dispositivo móvil, la base de datos y el HMI en forma de aplicación móvil.

Se espera obtener un prototipo funcional capaz de recopilar información sensorial y enviarla a una base de datos en la nube capaz de procesarla para posteriormente entregarla al usuario final, demostrando así la validez del proyecto y su funcionamiento básico.

Palabras clave: La nube, Dispositivo, Sensores, Base de datos, Contaminación.

ABSTRACT

According to the World Health Organization, "it is estimated that 1.3 million people worldwide die in a year due to urban air pollution" (World Health Organization, 2018).

Environmental pollution is a factor that has climbed according to the economic models that are currently handled, that is, mass production. In addition, transportation in the city also directly affects air quality; for the year 2015 "in the community 14 (El Poblado) were rolling 1,252 vehicles for every 1,000 inhabitants" (Álvarez, 2015), that is, there was approximately one car for every three people in that sector.

A study conducted from 1990 to 2012 reveals that "in Colombia, 258 million tons of gas emissions were emitted directly into the atmosphere as a result of human activities. The sectors that generated most emissions were the forest with 36%, the agricultural sector with 26%, the transport sector with 11%, the manufacturing industry with 11% and the mining and energy sector with 10% "(Correa, 2016).

There are already devices in charge of monitoring climatic variables related to air quality in the Aburrá valley. Therefore, a device is required that sends the information collected to the cloud to be processed and, unlike the other devices that are currently available, display results on a map of the city, indicating to the user in real time the quality of the air in each area.

As it is an application dedicated to cyclists, the device also has temperature and humidity sensors and a GPS. Additionally, the data network in the cloud is being updated with each route made by any individual that carries the device. For this reason, there is always updated data on pollution levels in different areas of the city.

In this project the Ulrich methodology is used to guide the development of the project. This will be carried out in three (3) main stages, corresponding to the specific objectives of the project. These stages refer to the development of the mobile device, the database and the HMI in the form of a mobile application.

It is expected to obtain a functional prototype able to collect sensory information and send it to a database in the cloud capable of processing it to later deliver it to the end user, thus demonstrating the validity of the project and its basic operation.

Keywords: The cloud, Sensors, Devices, Database, Contamination.

INTRODUCCIÓN

El planeta se está viendo afectado debido a actividades humanas como los desechos quemados en las fábricas, desechos, la tala de bosques, la destrucción de muchos ecosistemas y las emisiones de gases que se producen por los motores de combustión interna utilizados para el transporte, “según la secretaría de Movilidad: 222 238 (203 072 particulares y 19 166 taxis)” (Villa, 2017), donde para cada carro dando un recorrido de 8 km al día, se calcula una huella de carbono promedio de 4.8 toneladas de CO₂ por año.

A pesar de que en los gobiernos ya se está tomando conciencia de la situación en la que se encuentra el planeta, para el año 2017 “el 9,2 % del total de muertes naturales en el Aburrá es atribuible a la contaminación atmosférica” (Mario, 2017), teniendo en cuenta que los ciclistas están mucho más expuestos a la contaminación generada por las emisiones de gases de los vehículos; “en los pulmones de un ciclista entra hasta tres veces más aire contaminado que en los de un peatón” (Gregori, 2015), Es importante tomar acciones para la salud de ellos.

“Según un estudio realizado por la Universidad EAN, los niveles a los que se enfrentan los ciclistas por contaminación de Material Particulado Fino (PM 2.5) estuvieron en un rango entre los 70 y 135 µg/m³ para los días entre semana, y entre los 28 y 70 µg/m³ para un día de fin de semana” (Malaver, 2016). Para la organización mundial de la salud, “el nivel más bajo con los cuales se ha demostrado, con más del 95% de confianza, que la mortalidad total, cardiopulmonar y por cáncer de pulmón, aumenta en respuesta a la exposición prolongada al MP_{2,5} es de 10 µg/m³” (Organización Mundial de la Salud, 2005). Es decir que los ciclistas se están exponiendo a una contaminación en la cual está demostrado que es perjudicial para la salud.

En el presente trabajo de grado se desarrollará el prototipo de un dispositivo portátil, acoplable a una bicicleta, que le indique a quién lo está utilizando, la calidad del aire que le rodea. Se hará mostrando la información recolectada, procesada y analizada, a través de una aplicación móvil visualizando un mapa de tal forma que sea de fácil entendimiento para los usuarios.

1. PRELIMINARES

1.1 PLANTEAMIENTO DEL PROBLEMA

La contaminación afecta directamente la calidad de vida humana, un ex guía de ciclismo urbano dice: “los contaminantes químicos que se encuentran en el aire pueden causar sequedad de las mucosas, irritación y comezón en la piel (alergias) y en las vías respiratorias. También incide en el aumento de enfermedades vasculares y cardíacas, en la disminución de la capacidad de la sangre para transportar sustancias nutritivas y oxígeno al organismo” (Andunche, 2014).

Además, se ha comprobado que la calidad del aire influye en el estado de ánimo de las personas, según los estudios del doctor Joseph G. Allen “Los niveles de dióxido de carbono y compuestos orgánicos volátiles que comúnmente encontramos en edificios de oficinas convencionales se asocian con una disminución del rendimiento de los trabajadores en comparación a cuando esos mismos trabajadores se encuentran en entornos de construcción verde” (Sanz León, 2015).

Conociendo las condiciones en las que se encuentra el aire, también se pueden llegar a prevenir desastres naturales, contaminaciones tan altas que ya empiezan a afectar directamente a una población, es una alerta constante que nos recuerda el estado en el que está el ambiente que nos rodea y se sabe de qué es de lo que se están llenando los pulmones con cada respiración que se hace cada día (20 respiraciones por minuto según (Lindh, Pooler, Tamparo, Dahl, & Morris, 2013)).

Dependiendo del motivo por el cual se utiliza una bicicleta, ya sea por hobby, medio de transporte o por competir, hay mayor exposición a la contaminación ambiental. Una persona que va a su trabajo ubicado en el centro de la ciudad está más expuesta que alguien que se dedica a montar bicicleta en la montaña. Por lo tanto, es muy útil conocer las condiciones de los lugares por los que se hace el recorrido en bicicleta para tomar algunas medidas de precaución para la salud. Las condiciones hacen referencia a variables como temperatura, humedad, porcentaje de carbono, porcentaje de ozono, partículas PM 2,5 etc.

Camila Moncada es estudiante de física y va a la Universidad Nacional en bicicleta aproximadamente tres o cinco días a la semana. Ella dice: “Cuando uno está montando bicicleta, uno conduce al lado de todos los carros y se siente el humo que sale de ellos y de los carros. Por esto he empezado a tener más gripa y tos. También, mi piel se ha visto afectada porque siempre está muy sucia. Después de montar, uno siente que toda la mugre se le queda en la cara”. El testimonio de Camila comprueba una de las tesis de un estudio de la Universidad Nacional en el que se afirma que las personas que se movilizan en bicicleta son las más afectadas por las concentraciones de partículas ultrafinas en el ambiente (Arias, 2017).

Por lo tanto, surge la necesidad de desarrollar un dispositivo que mida y transmita los resultados de la calidad del aire y la temperatura. La información se sube a la nube para su procesamiento y almacenamiento. Adicionalmente, los resultados guardados en la base de datos serán visualizados en la interfaz.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Desarrollar el prototipo de un dispositivo portátil, acoplable a una bicicleta, que mida variables climatológicas y parámetros de distancia que serán procesados en la nube para visualizarse en un dispositivo móvil.

1.2.2 Objetivos Específicos

- Identificar variables ambientales relacionadas con actividades deportivas.
- Diseñar un dispositivo que recopile datos de cada uno de los sensores y los envíe a una base de datos inalámbricamente.
- Construir una aplicación móvil y una base de datos para la manipulación de las variables medidas por los sensores.
- Verificar los resultados obtenidos utilizando datos verídicos de centrales climatológicas.

1.3 MARCO DE REFERENCIA

1.3.1 Antecedentes

“Un grupo de estudiantes, tecnólogos, diseñadores, arquitectos de Nueva York y Ámsterdam, crearon un proyecto llamado “Air Quality Egg”, que es un sistema de sensores que permite a los usuarios monitorear de manera muy precisa al salir de sus casas los datos sobre CO₂ (dióxido de carbono) y NO₂ (dióxido de nitrógeno), que son los principales y más tóxicos contaminantes de la atmósfera, y poder compartir los resultados de este monitoreo vía internet con más usuarios.” (Gonzalez, 2012)

La empresa Plume Labs presentó un producto llamado Flow. Éste es un rastreador inteligente con el cual se puede medir la calidad de aire del lugar donde te encuentres, flow trabaja en conjunto con una aplicación complementaria que permite tener alertas personalizadas, saber en qué lugares cercanos hay aire puro o también te avisa en donde están los lugares más contaminados (Botero, 2017). Según el anuncio de Plume Labs, “uno de los objetivos de este producto es crear un movimiento global de monitores ciudadanos”

por medio de la app, pues los usuarios pueden ayudar a establecer mapas de la ciudad para encontrar ubicaciones de aire limpio”.

De acuerdo con lo que dice Fernando Benito, “Atmotube es un pequeño dispositivo portátil que puede medir la contaminación allá donde estés para proteger tu salud” (Benito, 2015). Es capaz de medir tanto el dióxido de carbono como otros agentes químicos presentes en el aire, además de realizar una sencilla comparación de los resultados a través de una aplicación para dispositivos iOS y Android que permite obtener la puntuación del aire que se respira con la siguiente clasificación: bueno, moderado, contaminado o muy contaminado.

También cuenta con una aplicación que muestra un mapa mundial sobre el nivel de contaminación del aire en distintos puntos geográficos, actualizado gracias a las mediciones tomadas por cada uno de los usuarios de este práctico medidor de polución.

1.3.2 Marco Teórico

Internet of the things (IOT)

“Internet de las cosas” por sus siglas en inglés, es la filosofía tecnológica de la interconexión informática de dispositivos. “IOT representa la próxima evolución de Internet, y que supondrá un avance enorme en su capacidad para recopilar, analizar y distribuir datos que se pueden convertir en información, en conocimiento y, en última instancia, en sabiduría” (Evans, 2011).

Calidad de aire

En el libro de la ley 7/94 de protección ambiental y sus reglamentos expresan lo siguiente, “se entiende por calidad del aire la adecuación a niveles de contaminación atmosférica, cualesquiera que sean las causas que la produzcan, que garanticen que las materias o formas de energía, incluidos los posibles ruidos y vibraciones, presentes en el aire no impliquen molestia grave, riesgo o daño inmediato o diferido, para las personas y para los bienes de cualquier naturaleza” (Junta de Andalucía, 1996).

App

Una aplicación móvil es un programa con interfaz gráfica que le muestra al usuario algún tipo de información o realiza algún tipo de proceso. En la actualidad existen apps para muchos tipos de actividades.

Tecnología de la información (TIC)

Las TIC se desarrollan a partir de los avances científicos producidos en los ámbitos de la informática y las telecomunicaciones. “Las TIC son el conjunto de tecnologías que permiten el acceso, producción, tratamiento y comunicación de información presentada en diferentes códigos (texto, imagen, sonido)” (Belloch, 2006).

Computación paralela

“El procesamiento paralelo es un tipo de procesamiento de la información, que permite que se ejecuten varios procesos concurrentemente [5, 10, 17, 30, 35]. El procesamiento paralelo puede ser de diferentes tipos: i) Un tipo es ejecutar procesos independientes simultáneamente, los cuales son controlados por el sistema operativo (usando tiempo compartido, multiprogramación y multiprocesamiento). ii) Otro tipo es descomponer los programas en tareas (controladas por el sistema operativo, los compiladores, los lenguajes de programación, etc.), algunas de las cuales pueden ser ejecutadas en paralelo. iii) Finalmente, el último tipo se basa en usar técnicas de encauzamiento para introducir paralelismo a nivel de instrucciones, lo que implica dividirlas en pasos sucesivos que pueden ser ejecutados en paralelo, cada uno procesando datos diferentes” (Aguilar & Leiss, 2004).

Computación en la nube

“Es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables (Servicios a través de internet), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios” (Mell & Grance, 2009).

Campos de aplicación en IOT:

- **SmartHome:** “El objetivo de la casa inteligente es el de proporcionarnos una mejor calidad de vida aumentando la comodidad, la seguridad y el confort, consiguiendo a su vez un mayor número de servicios de gran utilidad disponibles para el habitante de la casa” (Rodríguez & Fernández, 2009).
- **Smartcity:** El objetivo es tener la mayor cantidad de datos para facilitar la vida del ciudadano.
- **Agricultura inteligente:** Sus aplicaciones van desde monitorizar los cultivos, herramientas de soporte para la toma de decisiones, controlar automáticamente riego, protección de heladas, fertilización (Loureiro, 2015).
- **Wearables:** son dispositivos de consumo, entre ellos están pulseras para monitorizar, relojes inteligentes, que pueden la cantidad de pasos que se dan al día, las calorías quemadas, las distancias recorridas, etc.

2. METODOLOGÍA

Se utilizará como base la metodología Ulrich para el desarrollo del proyecto.

2.1 Identificación de los requerimientos del sistema a desarrollar

Se establecen cuáles son las variables que se deben medir para tener criterio y poder definir la calidad del aire. Dependiendo de las variables escogidas, se procede a hacer un análisis de la sensibilidad con la que se necesita que los sensores midan y posteriormente se evalúan los costos del proyecto como tal.

2.2 Diseño de concepto

Se procede a construir la caja negra, la caja que cuenta con entradas y salidas del sistema únicamente. Posteriormente se hace la caja transparente que incluye el proceso de cada variable, junto con un recorrido por toda la caja del comportamiento de cada variable y su evolución durante cada uno de los procesos que tiene el sistema.

Para cada función se hace una lista de las mejores alternativas de solución, se contemplan las necesidades del cliente y con respecto a las especificaciones que se encontraron, se evalúan cada una de las alternativas de solución para escoger la mejor opción. Se comparan aspectos como precio, tamaño, peso, resistencia al desgaste, etc., utilizando una matriz morfológica.

2.3 Diseño del detalle

En esta sesión ya se tienen claras las dimensiones del proyecto, es decir medidas de alto, ancho y largo. Se tiene una idea de la forma en que va a quedar el dispositivo (geometría), se debe de conocer el material a utilizar y proceso de producción. Se deben definir todos los componentes del producto teniendo en cuenta su estética. En esta etapa se suele tener listo el CAD del dispositivo para posteriormente comenzar con su construcción.

2.4 Construcción del prototipo

Se compra todo lo necesario para el desarrollo del proyecto y se inicia con la construcción del dispositivo, se ensamblan todas las partes y se acoplan todos los subsistemas para la funcionalidad del proyecto.

2.5 Pruebas y refinamiento

Comienza la toma de los datos y se compara su desempeño con las especificaciones que se tenían al comienzo del proyecto. Se hace un análisis sometiendo al prototipo a condiciones en las que se va a desenvolver en su vida útil, en este caso se instalará el aparato en una bicicleta y se le hará un recorrido por puntos con presencia del Siata para

una recolección de datos. Finalmente se toma una decisión, el prototipo es funcional o definitivamente hay que mejorarlo. En caso de que toque mejorarlo se corrigen sus imperfecciones y se vuelve a someter a las pruebas hasta que cumpla con las condiciones iniciales.

3. DESARROLLO DEL PROYECTO

3.1 VIGILANCIA TECNOLÓGICA

Para definir que variables son apropiadas para ser medidas por el dispositivo, se hizo una investigación sobre calidad del aire mostrada a continuación, se encontraron cuáles son las variables más perjudiciales para el ser humano, cuales sensores serían los apropiados para medirlas y la sensibilidad que se tiene que manejar para hacer un análisis adecuado. Para definir cuáles serían las variables de interés, se utilizaron los conocimientos de estudiantes de ingeniería ambiental de la Universidad EIA junto con la opinión de Alejandro Acosta, un experto en el tema de calidad del aire, actualmente profesor de la universidad de Medellín y también docente de cátedra en la universidad EIA. Alejandro ya ha participado en proyectos semejantes, estuvo presente en el desarrollo de un dispositivo, en donde el objetivo también era la medición y el análisis de la calidad del aire. Finalmente, las variables de interés son:

- **Temperatura y Humedad:**

“Una de las funciones fundamentales del cuerpo es regular la temperatura, que debe ser semi-constante para controlar el equilibrio y el balance de las funciones orgánicas y celulares. Cuando se hace ciclismo durante tiempos prolongados, la mayoría de la energía producida por los músculos ejercitados se disuelve con el calor, subiendo la temperatura corporal y provocando un riesgo en el cuerpo en forma de deshidratación” (Perujo, 2016).

- **Ozono:**

La presencia de Ozono en el aire que se respira puede producir los siguientes efectos: “Falta de aire y dolor cuando se hace una aspiración profunda, causa tos, dolor e irritación en la garganta, inflama y daña las vías respiratorias, agrava dolencias pulmonares como asma, enfisema y bronquitis crónica, incrementa la frecuencia de los ataques de asma y hace a los pulmones más susceptibles a infecciones” (González Aria, 2018).

- **Partículas Suspendidas:**

Las Partículas en suspensión en inglés Particulated matter (PM) “se clasifican según su medida y según cómo se comportan al respirarlas, más que según su contenido. Hay partículas de diámetro aerodinámico igual o inferior a 10 μm (PM10) que suelen llegar más allá de la garganta. Las que tienen un diámetro igual o inferior a 2,5 μm (PM 2,5) pueden llegar hasta los pulmones. Finalmente, las partículas ultrafinas, con un diámetro igual o inferior a 0,1 μm , pueden pasar de los alveolos pulmonares a la sangre” (Fundación Tierra, 2017).

Dependiendo del tamaño de la partícula, se determina que tan peligrosa es para el ser humano siendo las partículas ultrafinas las más peligrosas de todas. En este proyecto se trabajará con partículas PM 2,5.

- **Monóxido de carbono:**

El monóxido de carbono es un gas totalmente perjudicial para la salud, es invisible e inodoro y si se ingiere en altas cantidades puede acabar con la vida de una persona rápidamente. “Los bajos niveles de monóxido de carbono producen síntomas tales como dolores de cabeza leves, náuseas y dificultad para respirar. La exposición moderada al monóxido de carbono produce dolores de cabeza severos, mareos, confusión mental e incluso desmayos” (Osborne, 2017).

- **Dióxido de carbono:**

“El dióxido de carbono CO₂ es un gas presente en la atmosfera de forma natural en una concentración de 250 a 350 ppm, La concentración normal en un recinto cerrado esta entre 350 – 1000 ppm, Entre 1000 – 2000 ppm se considera calidad de aire bajo, de 2000 a 5000 ppm genera dolores de cabeza, insomnio y nauseas, a partir de 5000 ppm se genera una atmosfera toxica o deficiente de oxigeno con consecuencias fatales para el ser humano” (Mairal, 2013).

Para el 2 de mayo del 2013 la cantidad de CO₂ en el ambiente estaba en 399,50 ppm, la misma cantidad de CO₂ que hay en un recinto cerrado.

- **Compuestos volátiles orgánicos:**

“Los compuestos orgánicos volátiles (COVs) agrupan a una gran cantidad de sustancias químicas, que se convierten fácilmente en vapores o gases. Pueden tener altos grados de toxicidad. Además de sus efectos cancerígenos, la exposición a largo plazo a estos COVs puede causar lesiones de hígado, riñones y sistema nervioso central, mientras que a corto plazo puede causar irritación de los ojos y vías respiratorias, dolor de cabeza, mareos, trastornos visuales, fatiga, pérdida de coordinación, reacciones alérgicas de la piel, náuseas y trastornos de memoria” (Fernández, 2008).

Dado que el prototipo está enfocado para ciclistas, se necesita que el dispositivo tenga el menor tamaño posible, además de un costo asequible para cualquier persona promedio. Por lo tanto, para el almacenamiento, el acondicionamiento y el envío de la información de las variables mencionadas anteriormente, se descartan soluciones de altos costos y de medidas superiores a 20 centímetros por comodidad del usuario.

3.2 IDENTIFICACIÓN DE REQUERIMIENTOS

Los requerimientos y/o necesidades que se tienen para el dispositivo en el desarrollo del proyecto y su nivel de importancia están descritos en la tabla 1 a continuación. Dependiendo de estos requerimientos se procede a evaluar las alternativas de solución para cada función que se tiene.

Tabla 1.

Lista de requerimientos para el dispositivo.

N°	Necesidades	Nivel de Importancia (1 – 5)
1	El dispositivo sea portable	4
2	Fácil manejo	5
3	Alta resistencia a la intemperie	5
4	Fácil mantenimiento	4
5	Resistente a las variables que se van a medir	5
6	Vida útil del dispositivo	4
7	Resistencia a golpes y caídas	5
8	Interfaz amigable para el usuario	4
9	Estética	3
10	Bajo costo	5
11	Procesamiento en tiempo real	5
12	Protección de la información	5

3.2.1 Comparación con soluciones existentes

Actualmente, se encuentran en el mercado dispositivos con características similares a lo que acá se quiere implementar, éstos son los siguientes:

- **Sensor móvil de bajo costo:** Es un sensor muy pequeño diseñado por Unloquer, de costos más bajos que los sensores que se utilizan actualmente. El dispositivo utiliza un sensor chino que les permite medir el material particulado 2.5, además de variables tales como humedad y temperatura. Este aparato puede funcionar con una batería o conectado en casa con un cargador de celular. Los creadores decidieron no patentar el dispositivo, el sensor es de código abierto y la idea es utilizar componentes de bajo costo para que el producto total sea muy económico. El dispositivo funciona mientras que una persona lo lleva en el bolsillo, una moto o un automóvil y muestra que tan contaminada esta la zona en una escala de colores que va desde el verde (menos contaminación), pasa por amarillo, naranja y finalmente termina en rojo (mayor contaminación) como lo especificó diego Zambrano Benavides en su artículo (Zambrano, 2018).
- **The Airbeam:** Este dispositivo mide material particulado 2.5 y a cada segundo envía los datos recopilados por bluetooth a una aplicación de Android llamada Aircasting, donde son mapeados y graficados. Luego de pasar por la aplicación, la información es enviada a una página de internet de Aircasting donde se compara con las publicaciones de otros usuarios utilizando la misma aplicación. También cuenta con la escala de colores de verde hasta rojo, pasando por amarillo y naranja. “Además de crear reportes sobre la calidad del aire, también permite registrar lecturas de sonido y sus decibeles lo cual da un reporte más integral sobre la contaminación del ambiente” (Santoyo, 2014).
- **The Air Quality Egg:** “Es un sistema de sensores que permite monitorear datos sobre CO2 y NO2 para después compartirlo vía internet.” (Gonzalez, 2012). Este dispositivo tiene forma de huevo como su nombre lo indica, su propietario puede inscribir este aparato a la página web airquality.com para mapear los datos obtenidos.

“Lo mejor del Air Quality Egg es que lo puede utilizar cualquier persona en cualquier lugar, a diferencia de los datos que suministra el gobierno donde se manipula información de toda la región, esto no es muy útil para alguien que quiere conocer la situación actual afuera de su casa” (Treacy, 2013).

3.2.2 Especificaciones objetivo

Para el dispositivo se quieren cumplir los objetivos hasta un determinado punto, para este prototipo funcional se aceptarán hasta este punto como se puede observar en la tabla 2.

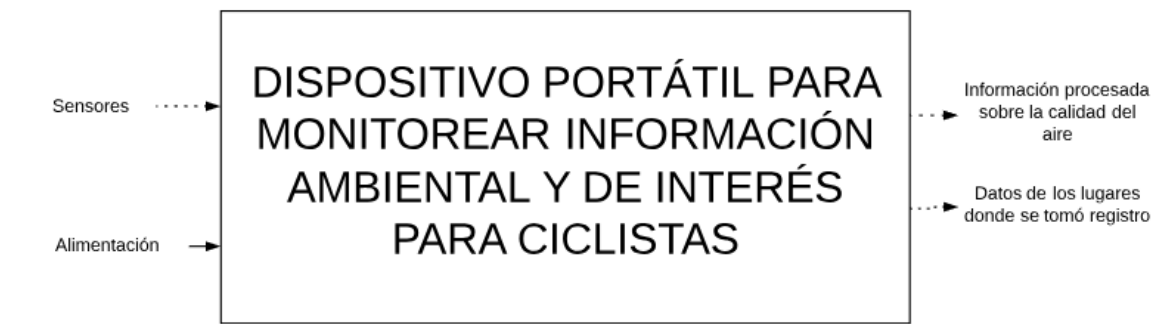
Tabla 2.
Especificaciones Objetivo

Medidas	Deseado	Aceptado	Unidades
Lectura de sensores en tiempo real	Si	Si	Binario
Reflejar los datos de contaminación en un mapa geográfico	Si	Si	Binario
Resistencia a la intemperie	Si	Si	Ppm
Tamaño del dispositivo	Muy pequeño	pequeño	Subjetivo
Tiempo de muestreo	60	90	Segundos
Indicadores de contaminación alta o baja	Si	Si	Binario
Duración de la batería	12	8	Horas

3.3 DISEÑO Y SELECCIÓN DE CONCEPTO

Para el diseño y selección de concepto se proceden los pasos a seguir planteados en la metodología Ulrich.

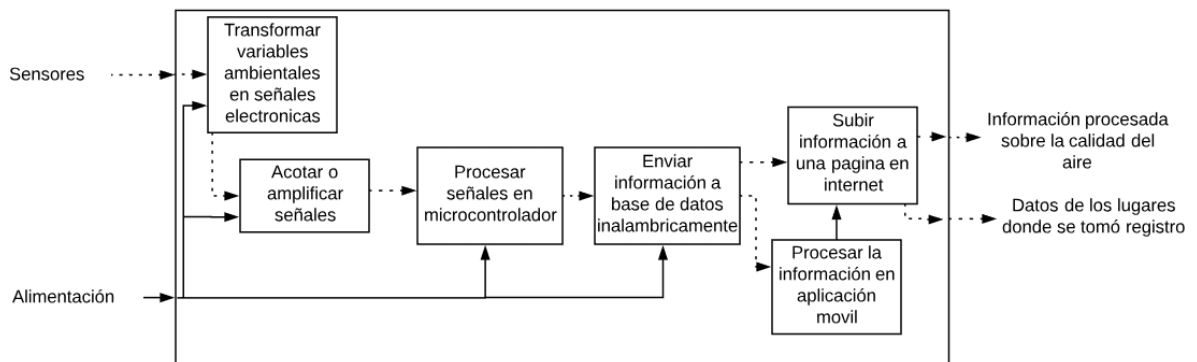
El primer paso es la caja negra, la cual está conformada solamente de las entradas y salidas del sistema, siendo el sistema el dispositivo que mide calidad de aire.



(Autores)

Figura 1. Caja Negra

Posteriormente se tiene la caja transparente, aquí se hace un análisis más profundo del sistema, dentro de ésta se tienen todas las funciones que tiene el dispositivo y además se puede observar cómo se van transformando las variables de entrada a medida que van pasando por cada una de las funciones hasta llegar a la salida.



(Autores)

Figura 2. Caja Transparente

3.3.1 Registro de soluciones

Para cada función se tienen varias alternativas de solución, las funciones son las siguientes:

Transformar variables físicas en señales electrónicas:

Para cumplir con esta tarea es necesario utilizar sensores, para este proyecto se tienen las siguientes variables y para cada variable se tienen estas opciones de sensores como se ilustra de la tabla 3 hasta la tabla 8.

Tabla 3.
Temperatura y Humedad



Alternativa de Solución	Definición	Ventajas	Desventajas
Módulo AM2302  (konnected, 2018)	Dispositivo que contiene sensor de humedad capacitiva y un termistor	Bajo costo No necesita acondicionamiento de señal Buena Documentación	Tiempo entre lecturas debe ser de 2 segundos
Sensor S52100  (Ammonit, 2017)	Dispositivo robusto que mide Humedad y temperatura de forma precisa en aire y otros gases no agresivos	Robusto Alta precisión Alta exactitud	Alto consumo de potencia Alto costo Requiere periodo de aclimatización inicial

Tabla 4.
Ozono:



Alternativa de Solución	Definición	Ventajas	Desventajas
Sensor MQ 131  (Winsen,2015)	detección de sensor de gas ozono O3	Más pequeño Liviano	Compleja implementación Requiere periodo de aclimatización inicial
Módulo MQ-131  (eVoltaPC,2018)	Módulo de detección de sensor de gas ozono O3	Menos probabilidad de fallo Fácil acople al dispositivo Buena Documentación	Ocupa más espacio que solo el sensor Requiere periodo de aclimatización inicial

Tabla 5.
Partículas




Alternativa de Solución	Definición	Ventajas	Desventajas
PPD42NS  (Tan,2014)	Sensor que utiliza un haz de luz para contar el número de partículas 2.5 en un volumen determinado aire	Ligero Bajo costo Buena Documentación Antecedentes	Ocupa mucho espacio Requiere Acondicionamiento empírico
GP2Y1010AU0F  (Bricotronica,2014)	Este sensor detecta pequeñas partículas de polvo mediante la emisión de un haz infrarrojo y la medida de su difusión a través del aire mediante un fototransistor	Consume poca corriente Buena sensibilidad	Ocupa mucho espacio Mucho ruido puede distorsionar la salida del sensor
PMS7003  (Newegg,2017)	Es un sensor que utiliza el principio de dispersión laser para contar partículas suspendidas en el aire	Distingue partículas de 0.3 micrómetros Rendimiento anti interferencia Muy Delgado	Ocupa mucho espacio Alto consumo de energía

Tabla 6.
Monóxido de carbono



Alternativa de Solución	Definición	Ventajas	Desventajas
Sensor MQ-7  (Windson,2014)	Sensor para medir concentración de monóxido de carbono en el aire	Más pequeño Liviano	Compleja implementación Requiere periodo de aclimatación inicial
Módulo MQ-7  (Teslabem,2017)	Módulo para medir concentración de monóxido de carbono en el aire	Menos probabilidad de fallo Fácil acople al dispositivo Buena Documentación	Ocupa más espacio que solo el sensor Requiere periodo de aclimatación inicial

Tabla 7.
Dióxido de carbono






Alternativa de Solución	Definición	Ventajas	Desventajas
MG-811  (Ardobot,2018)	Módulo para medir concentración de monóxido de carbono en el aire	Ligero Bajo costo Buena Documentación	Distingue partículas de 1 micrómetros Requiere periodo de aclimatación inicial
IAQ0001  (CO2Meter,2018)	Modulo para medir concentración de monóxido de carbono en el aire	Ligero	Requiere periodo de aclimatación inicial Poca documentación

Tabla 8.
Compuestos volátiles orgánicos (COV)

Alternativa de Solución	Definición	Ventajas	Desventajas
CCS811  (Adafruit,2017)	Sensor de gas digital para detectar un amplio rango de compuestos orgánicos volátiles totales	Bajo consumo de energía Pequeño Robusto ante altas temperatura	Requiere circuito extra para acoplar al dispositivo Requiere periodo de aclimatación inicial Costoso
MS1100  (ICStation,2017)	Sensor utilizado principalmente para detectar formaldehído, tolueno, benceno y otros gases de COV	Alta sensibilidad Ligero configurable	Bajo nivel de señal Requiere periodo de aclimatación inicial Costoso
MQ138  (Labelectronica,2017)	Módulo utilizado para detectar COV, entre ellos Tolueno, Acetona, Formaldehído y otros contaminantes	Buena Documentación Ligero Sensibilidad configurable	Requiere periodo de aclimatación inicial Costoso

En las tablas 9,10 y 11 se encuentran las alternativas de solución para la base de datos, la transmisión de los datos y para el procesamiento de los datos.

Tabla 9.

Almacenar en base de datos:






Alternativa de Solución	Definición	Ventajas	Desventajas
<p>MySQL</p>  <p>(Oracle, s.f.)</p>	Es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL)	<p>Open Source</p> <p>Excelente rendimiento</p> <p>Bajo costo computacional</p>	<p>Limitaciones en el procesamiento de las búsquedas</p> <p>No sirve para bases de datos que requieren alta disponibilidad (ambiente corporativo)</p>
<p>MongoDB</p>  <p>(Baer,2018)</p>	Es una base de datos NoSQL de código abierto orientado a documentos con un alto rendimiento y disponibilidad.	<p>Open Source</p> <p>Alto rendimiento</p> <p>Alta disponibilidad</p> <p>Escalable (Bases de datos muy grandes)</p>	<p>Procesos lentos y complejos</p> <p>Poca fiabilidad</p> <p>Poca consistencia</p>
<p>Oracle DB</p>  <p>DATABASE (Smeets,2007)</p>	Es un lenguaje de computación diseñado para gestionar datos almacenados en bases de datos relacionales	<p>Motor de base de datos objeto-relacional más usado a nivel mundial</p> <p>Eficiente</p> <p>Estandarización</p>	<p>Las licencias del software son muy costosas</p> <p>Complejo</p> <p>Funcionalidad limitada</p>

Tabla 10.

Enviar y recibir información

Alternativa de Solución	Definición	Ventajas	Desventajas
<p>Bluetooth</p>  <p>(i+D, s.f.)</p>	Es una tecnología de ondas de radio de corto alcance (2.4 GHZ de frecuencia)	<p>Servicio gratis</p> <p>Buena velocidad de transmisión</p> <p>Baja complejidad</p>	<p>Seguridad</p> <p>Reducido alcance</p>
<p>WI-Fi</p>  <p>(Pixabay,2017)</p>	Un tipo de conexión que puede unir un sinnúmero de dispositivos de diversas características	<p>Conexión en cualquier lugar</p> <p>Elección entre señales libres o seguras</p> <p>Conectividad inalámbrica</p>	<p>Es afectada por la radiación electromagnética</p> <p>Seguridad</p> <p>Distancia limitada para recibir la señal</p>





<p>GPRS – GPS</p>  <p>(Electrodragon,2013)</p>	<p>sistema que permite mandar y recibir paquetes de datos usando una red de telefonía por satélite</p>	<p>Alta eficiencia</p> <p>Cambio de velocidad de transmisión</p> <p>Manejo de la congestión de los datos</p>	<p>Colas de espera</p> <p>Congestión de los canales por donde se envían los datos</p>
---	--	--	---























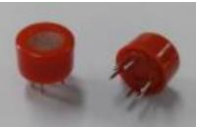






Tabla 11.
Procesamiento de datos













Alternativa de Solución	Definición	Ventajas	Desventajas
<p>Arduino/ Atmega 328P</p>  <p>(researchdesignlab, 2012)</p>	<p>Es un Circuito integrado de alto rendimiento el cual será programado utilizando arduino</p>	<p>Económico</p> <p>Arduino como lenguaje de programación</p> <p>Pequeño</p>	<p>Necesidad de utilizar bootloader</p> <p>Sensibles al ruido</p>
<p>Raspberry</p>  <p>(Samm,2015)</p>	<p>Es un microcomputador solo con los componentes necesarios para su funcionamiento básico</p>	<p>Muchas funciones</p> <p>Fácil programación</p> <p>Velocidad del procesador</p>	<p>Costoso</p> <p>Gran tamaño</p> <p>Alto consumo de potencia</p>
<p>PIC16F877A</p>  <p>(Microchip,2003)</p>	<p>Dispositivo programable capaz de realizar diferentes actividades que requieran del procesamiento de datos entre diferentes dispositivos</p>	<p>Económico</p> <p>Pequeño</p> <p>Eficiente</p>	<p>Compleja programación</p> <p>Sensibles al ruido</p> <p>Instrucciones reducidas</p>

3.3.2 Matriz Morfológica

Para tener una mejor visión de las alternativas propuestas y poder hacer una buena selección de cuál será la mejor solución, se hace uso de la matriz morfológica.

Tabla 12.
Matriz Morfológica

Función		Alternativas de solución		
		1. 	2. 	3. 
Transformación de variables físicas en señales electrónicas	Temperatura y Humedad		Módulo AM2302	  Sensor S52100
		 (konnected, 2018)	 (Ammonit, 2017) (Speedy Tech, 2016)	
Transformación de variables físicas en señales electrónicas	Ozono		Sensor MQ 131	  Módulo MQ-131
		 (Winsen,2015)	 (eVoltaPC,2018)	
Transformación de variables físicas en señales electrónicas	Partículas		PPD42NS	  GP2Y1010AU0F
		 (Tan,2014)	 (Bricotronica,2014)	 (Newegg,2017)
Transformación de variables físicas en señales electrónicas	Monóxido de carbono		Sensor MQ-7	  Módulo MQ-7
		 (Windson,2014)	 (Teslabem,2017)	
Transformación de variables físicas en señales electrónicas	Dióxido de carbono	 	MG-811	
		 (Ardbot,2018)	 (CO2Meter,2018)	

Transformación de variables físicas en señales electrónicas	Compuestos orgánicos volátiles	 <p>CCS811</p> <p>(Adafruit,2017)</p>	 <p>MS1100</p> <p>(ICStation,2017)</p>	 <p>MQ138</p> <p>(Labelectronica,2017)</p>
Almacenar en base de datos		 <p>MySQL</p> <p>(Oracle, s.f.)</p>	 <p>MongoDB</p> <p>(Baer,2018)</p>	 <p>Oracle DB</p> <p>(Smeets,2007)</p>
Enviar y recibir datos		 <p>Bluetooth</p> <p>(i+D, s.f.)</p>	 <p>Wi-Fi</p> <p>(Pixabay,2017)</p>	 <p>GPRS – GPS</p> <p>(Electrodragon,2013)</p>
Procesamiento de datos		 <p>Arduino/Atmel</p> <p>(researchdesignlab, 2012)</p>	 <p>Raspberry</p> <p>(Samm,2015)</p>	 <p>PIC16F877A</p> <p>(Microchip,2003)</p>

Solución 1.

Esta solución propone para el diseño del circuito solo los sensores, es una alternativa más económica, pero requiere de un trabajo extra que incluye el acople de los sensores al circuito principal. Esta solución propone una base de datos tipo MongoDB, es una base de datos No-SQL, no es relacional. Estas bases de datos son más flexibles en cuanto a consistencia de datos y se han convertido en una opción que intenta solucionar algunas limitaciones que tiene el modelo relacional. La comunicación por wifi es una solución económica y es muy utilizada actualmente, para el procesamiento se tiene un PIC16F877A,

un microcontrolador de poco volumen, liviano, de bajo costo, pero es un poco más engorrosa su programación.

Solución 2.

Se utilizan también los módulos con los sensores incluidos para minimizar cableado y disminuir probabilidades de errores por fallos en componentes o mala conexión entre dos terminales. Se propone una base de datos MySQL, una base de datos de código libre, con una interfaz sencilla, pero cumple con todo lo que se necesita para el proyecto. La transmisión de datos se hará directamente desde el GPRS simplificando el proceso y reduciendo el número de componentes a utilizar. Se utiliza Arduino como sistema de desarrollo empleando una tarjeta arduino nano y después se programa en un microcontrolador (Atmega 328P) utilizando boot loader.

Solución 3.

Esta estrategia reemplaza los sensores por módulos que incluyen los sensores, así se reduce el cableado, el uso de más componentes y así se disminuyen las probabilidades de error. Se utiliza una base de datos relacional de tipo Oracle, la transmisión se hace mediante módulos bluetooth, esta tecnología es incluso más económica que el Wifi, pero no cubre grandes distancias. Para el procesamiento de datos se tiene una Raspberry, cumple las mismas funciones que un Arduino, pero se necesitan unos cuantos detalles más para su programación.

3.3.3 Selección de concepto

Después de tener la matriz morfológica con sus posibles combinaciones, se analiza cada posibilidad teniendo en cuenta las especificaciones requeridas y su nivel de importancia en el proyecto.

El dispositivo final debe tener resistencia a golpes por el simple hecho de que va montado en una bicicleta y hay altas probabilidades de accidentes, choques o brusquedad del usuario para manipular la bicicleta. Dado lo anterior se le otorgará un 20% de importancia a la resistencia a golpes en el proyecto.

Otro factor determinante para la selección del concepto fue el costo, el dispositivo está enfocado para cualquier ciclista, desde adultos profesionales hasta adolescentes que simplemente se preocupan por su salud cuando salen a montar en bicicleta. Por lo tanto, el producto debe de llamar la atención por tener un costo asequible, consiguiendo un puntaje del 30% de importancia.

Para mayor eficiencia en el desarrollo del producto, es decir, para gastar la menor cantidad de recursos posibles (ya sea en mano de obra como en materiales), se le asigna un 20% de importancia a la simplicidad en el circuito y la programación.

Debido a que el producto funciona acoplado en una bicicleta, es importante tener en cuenta las longitudes máximas del prototipo para evitar que éste entorpezca la maniobrabilidad o comodidad del usuario. Por eso se le asigna un 30% de importancia a el tamaño del dispositivo, donde entre más pequeño sea, mejor calificación tendrá.

A continuación, se muestran todos los criterios junto con su % de importancia para cada una de las opciones en la Tabla 13.

Tabla 13.
Resultado selección concepto

Criterio	Porcentaje de importancia	Calificación Solución 1.	Calificación Solución 2.	Calificación Solución 3.
Bajos costos	20%	5	3	1
Tamaño	30%	3	5	1
Simplicidad circuito y programación	20%	1	5	3
Resistencia a golpes	30%	1	5	3
Puntuación Total	100%	2,4	4,6	2

De acuerdo a la tabla 13 se escoge el mejor conjunto de soluciones. En este caso se puede observar que la solución número 2 es la que mayor puntuación total tiene. por lo tanto, se selecciona para el desarrollo del proyecto.

En la tabla 14 se observan los costos para los sensores que se escogieron finalmente junto con su respectivo fabricante.

Tabla 14.
Información Sensores

Fabricante	Referencia	Precio (USD)
Ai thinker	Ai thinker A7	\$15
-	Mq131	\$14.88
Shinyei	Ppd42ns	\$10.33
-	Mq7	\$3.35
-	Mg811	\$25.11
-	Ccs811	\$20
-	Bmp180	\$2.41
Asair	Am2302	\$3.81

3.4 DISEÑO DE DETALLE

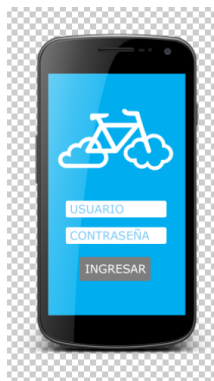
Teniendo en cuenta las soluciones escogidas en la selección del concepto, se diseña la página web, el circuito electrónico y el diseño de la estructura mecánica del dispositivo.

3.4.1 Página Web

- **Diseño Gráfico**

Se desarrolla la interfaz de tal forma que sea agradable para el cliente. Se escoge el color azul como fondo ya que aporta serenidad, paz, genera una sensación de seguridad y confianza en el usuario.

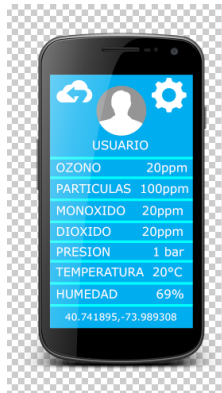
El diseño del logo fue realizado por el equipo de trabajo. Este logo busca enlazar el concepto de IoT (simbolizado mediante las nubes), con el ciclismo.



(Autores)

Figura 3. Interfaz Inicio de sesión inicial

La ventana de la información de los sensores también maneja el mismo fondo y se propone de la siguiente manera



(Autores)

Figura 4. Interfaz sensores inicial

En la figura 5, se muestra la contaminación en forma de mapa de calor utilizando los datos recopilados por los usuarios. En la zona inferior de la interfaz se muestra el estado del aire que indica el dispositivo utilizando el código de colores (verde = bien, amarillo = regular, rojo = mal)

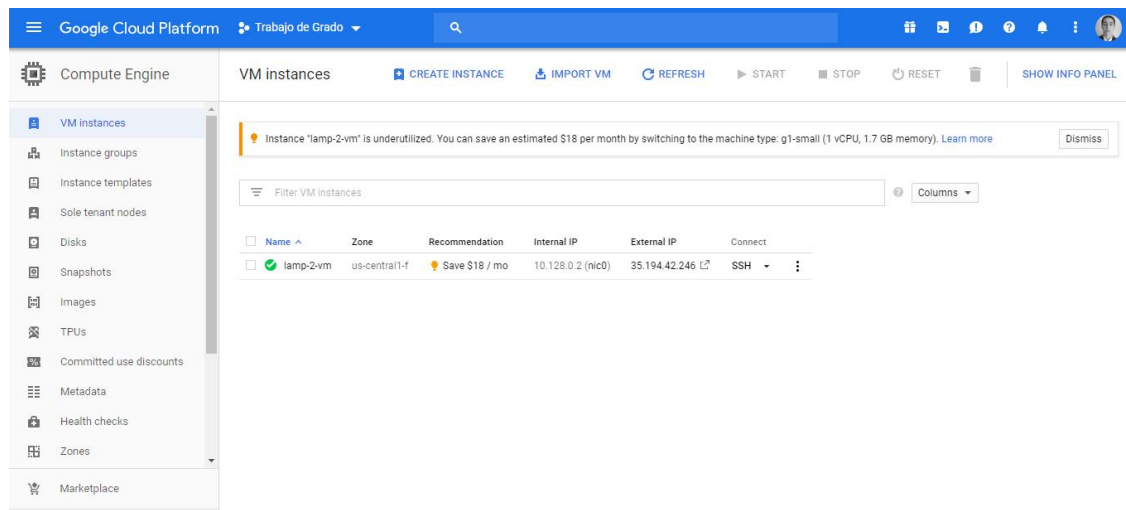


(Autores)

Figura 5. Interfaz mapa de calor inicial

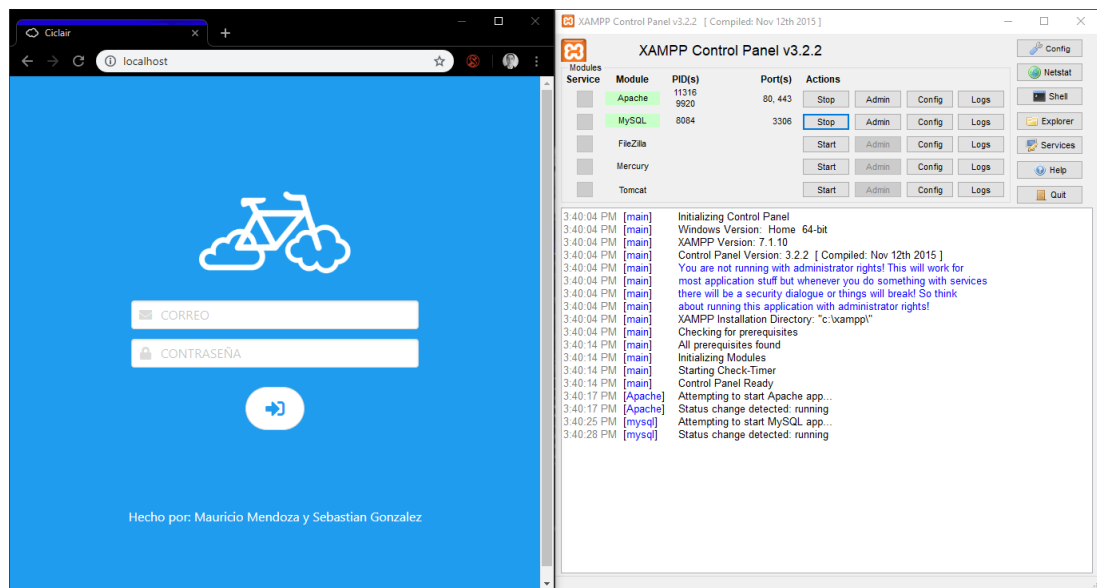
- **Desarrollo de la Pagina**

Se desarrolló en XAMP, luego se migró a gcloud donde se estableció una máquina virtual con un LAMP stack. XAMP es un software libre que cuenta con herramientas para el desarrollo de páginas web, estas herramientas son: gestión de bases de datos mysql, el servidor web apache, intérpretes para lenguajes de script php y Perl. En el proyecto se utilizan principalmente la gestión de bases de datos mysql y el servidor web apache.



(Autores)

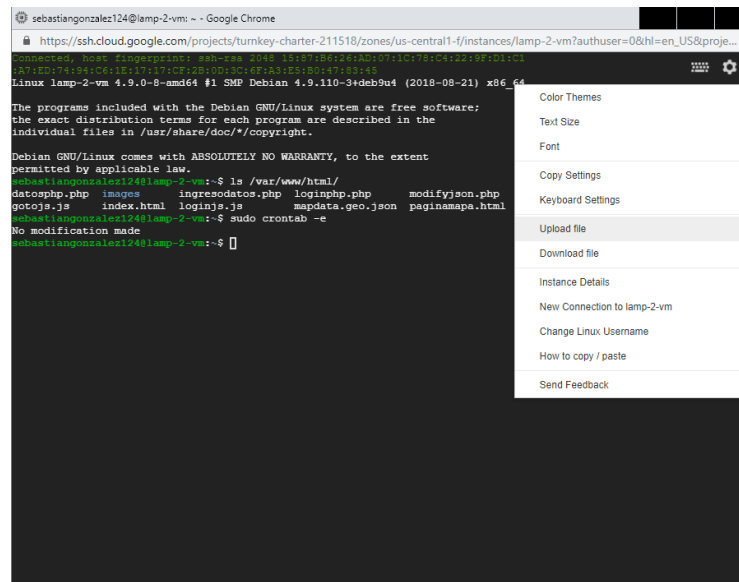
Figura 6. gcloud



(Autores)

Figura 7. XAMP

La configuración e interacción con dicha instancia se hace mediante SSH como se muestra a continuación.



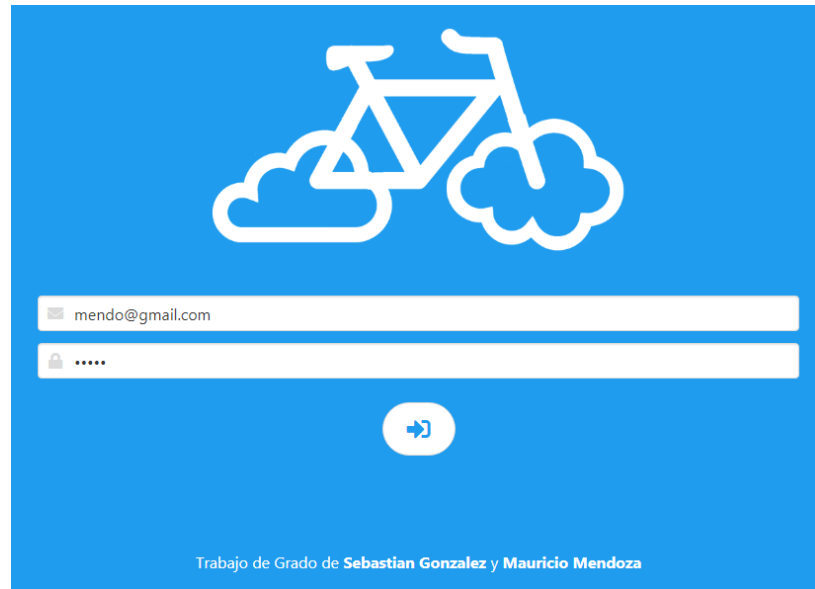
(Autores)

Figura 8. SSH

Al ser una máquina de Linux, se utilizan comandos propios del sistema operativo como LS (muestra una lista de los archivos dentro del destino) y MV (mueve un archivo de una carpeta a otra). Gracias a la interfaz propia de gcloud, se pueden subir archivos directamente a la máquina virtual, de esta manera fue que se migró la página web.

Durante el desarrollo se efectuaron cambios, unos por limitaciones de código, otros por simple estética. Después de que se realizaron los cambios, la página web quedó con 3 interfaces para la interacción con el usuario:

Inicio de Sesión: Cada Usuario registrado tendrá que ingresar a la página utilizando su correo y una contraseña.



(Autores)

Figura 9. Inicio de sesión

Para la estructura de esta página, se tiene el siguiente código HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.7.1/css/bulma.min.css">
  <link rel="shortcut icon" href="images/cloudicon.ico">
  <title>Ciclair</title>
  <script
    src="https://code.jquery.com/jquery-3.2.1.js"
    integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jcIzLy3oi35EouyE="
    crossorigin="anonymous">
  </script>
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.2.0/css/all.css" integrity="sha384-
  -hwVjflwFxL6sNzntih27bfxkr27PmbbK/iSvJ+a4+0owXq79v+1sFkw54bOGbiDQ" crossorigin="anonymous">
</head>
<body>
  <section class="section has-background-info is-bold"></section>
  <section class="section has-background-info is-bold">
    <div class="columns is-mobile">
      <div class="column is-one-third is-offset-one-third">
        <figure class="image">
          <a></a></a>
        </figure>
      </div>
    </div>
    <div class="columns is-mobile">
      <div class="column is-three-fifths is-offset-one-fifth">
        <p id="respuesta" class="is-size-4 has-text-centered has-text-danger"></p>
      </div>
    </div>
  </section>
</body>
```

(Autores)

Figura 10. Código *Index.html* parte 1

```

<div class="field">
  <p class="control has-icons-left">
    <input id="correo" class="input" type="email" placeholder="CORREO">
    <span class="icon is-small is-left">
      <i class="fas fa-envelope"></i>
    </span>
  </p>
</div>
<div class="field">
  <p class="control has-icons-left">
    <input id="contraseña" class="input" type="password" placeholder="CONTRASEÑA">
    <span class="icon is-small is-left">
      <i class="fas fa-lock"></i>
    </span>
  </p>
</div>
</div>
</div>
<div class="columns is-mobile">
  <div class="column is-one-fifth is-offset-two-fifths">
    <div class="field">
      <p class="control has-text-centered">
        <button id="boton" class="button is-large is-rounded has-background-white has-text-info">
          <i class="fas fa-sign-in-alt"></i>
        </button>
      </p>
    </div>
  </div>
</div>
</section>
<script src="loginjs.js"></script>
</body>

```

(Autores)

Figura 11. Código *Index.html* parte 2

```

<footer class="footer has-background-info is-bold">
  <div class="container">
    <div class="content has-text-centered has-text-white">
      <p>Hecho por: Mauricio Mendoza y Sebastian Gonzalez</p>
    </div>
  </div>
</footer>
</html>

```

(Autores)

Figura 12. Código *Index.html* parte 3

El html de las figuras 10, 11 y 12 (código *index.html*) dicta la estructura física del frontend de la página, tiene un logotipo, los campos de texto para el usuario y un botón, La estética de este HTML, se enriquece utilizando Bulma (framework de CSS).

Este archivo HTML llama a un archivo JavaScript llamado *loginjs.js* que se encarga de darle la funcionalidad a la página, ésta verifica que la información suministrada por el usuario tenga sentido, es decir, que no se dejen campos sin llenar y que el correo si tenga formato de correo electrónico, para identificar el formato de correo se está utilizando una función llamada *validEmail()*.

Código *loginjs.js* (archivo JavaScript):

```
function validEmail(v) {
    var r = new RegExp("[a-z0-9!#$%&'*/=?^_`{|}~-]+(?:\\.[a-z0-9!#$%&'*/=?^_`{|}~-]+)*@(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?)");
    return (v.match(r) == null) ? false : true;
}

$("#boton").click(function(){
    var correocontraseña={
        correo:$("#correo").val(),
        contraseña:$("#contraseña").val()
    };
    if (correocontraseña.correo=="" || correocontraseña.contraseña=="") {
        $("#respuesta").html("Campo vacio");
    }
    else{
        if(validEmail(correocontraseña.correo)){
            $.post("loginphp.php",correocontraseña,function(resp,status){
                //window.alert(resp)
                if(resp=="Bad"){
                    $("#respuesta").html("Clave errada");
                }
                else if(resp=="Null"){
                    $("#respuesta").html("Correo no registrado");
                }
                else{
                    window.location.href='paginamapa.html';
                }
            });
        }
        else{
            $("#respuesta").html("Correo no valido")
        }
    }
});
});
```

(Autores)

Figura 13. Código *loginjs.js* parte 1

Una vez se realiza la verificación y se verifica que todo esté en orden, se envía la información del usuario y la contraseña a un archivo PHP cuyo nombre es *loginphp.php*.

Loginphp.php se encarga de comparar con una base de datos en donde están todos los usuarios registrados y verifica que el nombre de usuario ingresado si existe y si la contraseña corresponde con ese nombre de usuario ingresado.

Código *loginphp.php* (archivo PHP):

```
<?php
$nombre = $contraseña = "";
if($_SERVER['REQUEST_METHOD'] == "POST") {
    $host = "localhost";
    $dbname = "trabajodegrado";
    $user = "root";
    $pass = "dEBek7hvEBpiMp";
    $conectado = false;
    try {
        $pdo = new PDO('mysql:dbname=.'.$dbname.';host='.$host,$user,$pass);
        $conectado = true;
    } catch(PDOException $e) {
        $conectado = false;
    }
    $nombre = htmlspecialchars($_POST["correo"]);
    $query = $pdo->prepare("SELECT Clave FROM usuarios WHERE Correo = \"\$nombre\"");
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_COLUMN);
    $contraseña = htmlspecialchars($_POST["contraseña"]);
    //echo implode(' ', $resultado);
    $query = $pdo->prepare("SELECT Usuario FROM usuarios WHERE Correo = \"\$nombre\"");
    $query->execute();
    $resultadoUsuario = $query->fetchAll(PDO::FETCH_COLUMN);
    if($resultado==Null){
        echo "Null";
    }
}
```

(Autores)

Figura 14. Código *loginphp.php* parte 1

```
    else{
        if($contraseña==$resultado[0]){
            session_start();
            $_SESSION['user'] = $resultadoUsuario[0];
            echo $_SESSION['user'];
        }
        else{
            echo "Bad";
        }
    }
}
if($_SERVER['REQUEST_METHOD'] == "GET"){
    session_start();
    echo $_SESSION['user'];
}
?>
```

(Autores)

Figura 15. Código *loginphp.php* parte 2

Este código utiliza peticiones para comparar al usuario ingresado con los usuarios guardados en la base de datos, en este caso se está utilizando MYSQL como sistema de manejo para la base de datos y se utiliza phpMyAdmin para administrar la base de datos de una forma más amigable para el administrador de red.

Si el usuario ingresado no existe, el programa devuelve “null” a *loginjs.js*. Si la clave no coincide con el usuario, el programa devuelve “bad” a *loginjs.js* y si el usuario existe y además la contraseña es la correcta devuelve un inicio de sesión a *loginjs.js*.

Si la respuesta es “null”, se envía un mensaje diciendo que el correo no está registrado, si recibe “bad”, dice que la clave esta errada y si recibe el inicio de sesión, pasa al próximo HTML llamado *paginamapa.html* en la sesión del usuario que ingresó.

Mapa y Estado actual: Una vez se ingresa al sitio web, se le suministra al usuario un mapa, indicándole el estado en el que se encuentra la calidad de aire en donde se encuentra el dispositivo. Esto se hace mediante un cuadro con escala de colores siendo rojo el peor, amarillo intermedio y verde el mejor.

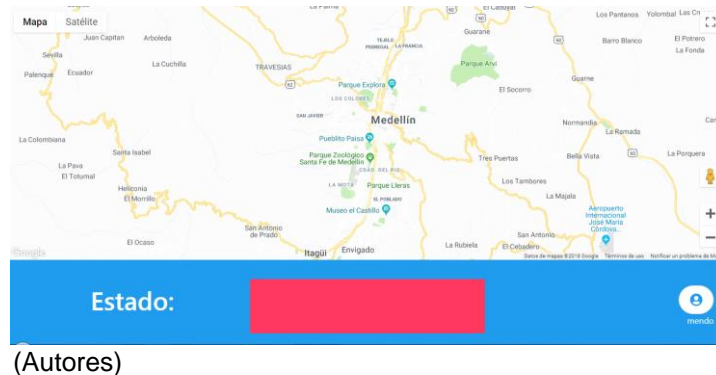


Figura 16. Mapa y estado actual del aire

Para la estructura de esta página, se tiene el siguiente el código HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    /* Always set the map height explicitly to define the size of the div element that contains the map. */
    #map {
      height: 75%;
    }
    /* Optional: Makes the sample page fill the window. */
    html, body {
      height: 100%;
      margin: 0;
      padding: 0;
    }
  </style>
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.7.1/css/bulma.min.css">
  <link rel="shortcut icon" href="images/cloudicon.ico">
  <title>Cicclair</title>
  <script
    src="https://code.jquery.com/jquery-3.2.1.js"
    integrity="sha256-DZAnKJ/6XZ9si04Hgrsxu/8s717jcIzLy3oi35EouyE="
    crossorigin="anonymous">
  </script>

  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.2.0/css/all.css" integrity="sha384-hWVjflwFxL6sNzN"
  </head>

  <body class="has-background-info is-bold">
    <div id="map"></div>
```

(Autores)

Figura 17. Código *paginamapa.html* parte 1

```

<section class="section has-background-info is-bold">
  <div class="columns is-mobile">
    <div class="column is-one-third">
      <p class="title is-size-2-mobile is-size-1-desktop has-text-white has-text-centered">C.A.</p>
    </div>
    <div id="estado" class="column is-one-third"></div>
    <div class="column is-one-third">
      <p class="has-text-right">
        <a id="botongotousuario" class="button is-large is-rounded">
          <span class="icon has-text-info">
            <i class="fas fa-user-circle"></i>
          </span>
        </a>
        <p id="ident" class="has-text-white has-text-right"></p>

        <p id="lat" class="has-text-white has-text-right"> 6.244203</p>
        <p id="lon" class="has-text-white has-text-right">-75.581215</p>
      </p>
    </div>
  </div>
</section>

<script src="gotojs.js"></script>
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCmuIGQgJ0o9sxFtY_iu880hPdM_F59JsY&callback=initMap"></script>
<script>
  var usuarioActual={
    user:"",
  };

```

(Autores)

Figura 18. Código *paginamapa.html* parte 2

```

$.get("loginphp.php","",function(resp,status){
  usuarioActual.user=resp;
  $("#ident").html(usuarioActual.user);
  //window.alert("dentro de get "+usuarioActual.user);

  var map;
  $.post("datosphp.php",usuarioActual,function(resp,status){
    var datos =JSON.parse(resp);
    //window.alert("POST")
    //window.alert(datos.INDICE);
    var longitud = datos.GPS.substring(0, datos.GPS.indexOf(','));
    var latitud = datos.GPS.substring(datos.GPS.indexOf(',')+1,datos.GPS.length);
    document.getElementById('lat').innerHTML = latitud;
    document.getElementById('lon').innerHTML = longitud;
    //$("#lat").html(latitud);
    //$("#lon").html(longitud);
    //window.alert("del PHP long:"+longitud+"lat:"+latitud);
    //window.alert("del HTML long:"+ parseFloat(document.getElementById('lon').innerHTML) +"lat:"+ parseFloat(document

    map = new google.maps.Map(document.getElementById('map'), {
      center: { lat: parseFloat(document.getElementById('lat').innerHTML), lng: parseFloat(document.getElementById('lon').innerHTML)},
      //center: { lat: latitud, lng: longitud},
      zoom: 14,
      styles: mapStyle,
      gestureHandling: 'greedy'
    });
  });

```

(Autores)

Figura 19. Código *paginamapa.html* parte 3

```

var user = new google.maps.Circle({
  strokeColor: '#3273dc',
  strokeOpacity: 1,
  strokeWeight: 5,
  fillColor: '#209cee',
  fillOpacity: 1,
  map: map,
  center: { lat: parseFloat(document.getElementById('lat').innerHTML), lng: parseFloat(document.getElementById('
  radius: 50
});

map.data.loadGeoJson('mapdata.geo.json');
map.data.setStyle(styleFeature);
var script = document.createElement('script');
script.setAttribute(
  'src',
  'mapdata.geo.json');
document.getElementsByTagName('head')[0].appendChild(script);

eqfeed_callback('mapdata.geo.json');
// Defines the callback function referenced in the jsonp file.

function eqfeed_callback(data) {
  map.data.addGeoJson(data);
}

function styleFeature(feature) {
  var low = [151, 83, 34]; // color of mag 1.0
  var high = [5, 69, 54]; // color of mag 6.0 and above
  var minMag = 0.0;
  var maxMag = 10.0;
  // fraction represents where the value sits between the min and max
  var fraction = (Math.min(feature.getProperty('mag'), maxMag) - minMag) /

```

(Autores)

Figura 20. Código *paginamapa.html* parte 4

```

    (maxMag - minMag);

    var color = interpolateHsl(low, high, fraction);
    return {
      icon: {
        path: google.maps.SymbolPath.CIRCLE,
        strokeWeight: 0.0,
        strokeColor: '#fff',
        fillColor: color,
        fillOpacity: 0.5,
        // while an exponent would technically be correct, quadratic looks nicer
        scale: 4*map.zoom/Math.pow(feature.getProperty('mag'), 2)
      },
      zIndex: Math.floor(feature.getProperty('mag'))
    };
  }

  function interpolateHsl(lowHsl, highHsl, fraction) {
    var color = [];
    for (var i = 0; i < 3; i++) {
      // Calculate color based on the fraction.
      color[i] = (highHsl[i] - lowHsl[i]) * fraction + lowHsl[i];
    }

    return 'hsl(' + color[0] + ',' + color[1] + '%,' + color[2] + '%)';
  }

  var mapStyle = [{
    'featureType': 'all',
    'elementType': 'all',
    'stylers': [{ 'visibility': 'on' }]
  }, {
    'featureType': 'landscape',
    'elementType': 'geometry',
    'stylers': [{ 'visibility': 'on' }, { 'color': '#fcfcfc' }]
  }

```

(Autores)

Figura 21. Código *paginamapa.html* parte 5

```

  }, {
    'featureType': 'water',
    'elementType': 'labels',
    'stylers': [{ 'visibility': 'on' }]
  }, {
    'featureType': 'water',
    'elementType': 'geometry',
    'stylers': [{ 'visibility': 'on' }, { 'hue': '#5f94ff' }, { 'lightness': 60 }]
  }
  ]];
});
</script>
</body>
</html>

```

(Autores)

Figura 22. Código *paginamapa.html* parte 6

Paginamapa.html utiliza el código de Google maps, que es un api público suministrado por Google en donde se adquiere una apikey para su uso.

Dentro de la estructura de google maps, se definen los parámetros para graficar los puntos de colores de donde se está tomando la información, para esto se llama a un archivo llamado *mapdata.geo.json*.

[illegible]

Figura 23. Archivo *mapdata.geo.json*

47

La información presentada en este documento es de exclusiva responsabilidad de los autores y no compromete a la EIA.

```

<?php
    $host = "localhost";
    $dbname = "trabajodegrado";
    $user = "root";
    $pass = "dEBek7hvEBpiMp";
    $conectado = false;
    try {
        $pdo = new PDO('mysql:dbname='.$dbname.';host='.$host,$user,$pass);
        $conectado = true;
    } catch(PDOException $e) {
        $conectado = false;
    }
    $query = $pdo->prepare("SELECT * FROM datos ORDER BY TIEMPO DESC LIMIT 50");
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    $geojson = array(
        'type' => 'FeatureCollection',
        'features' => array()
    );
    foreach ($resultado as $row){
        list($long, $lat) = explode(',', $row['GPS']);
        $feature = array(
            // 'id' => $row['usuario'],
            'type' => 'Feature',
            'geometry' => array(
                'type' => 'Point',
                # Pass Longitude and Latitude Columns here

                'coordinates' => array((float)$long, (float)$lat)
            ),
            # Pass other attribute columns here
            'properties' => array(
                'mag' => (float)$row['INDICE'],

```

(Autores)

Figura 24. Código *modifyjson.php* parte1

```

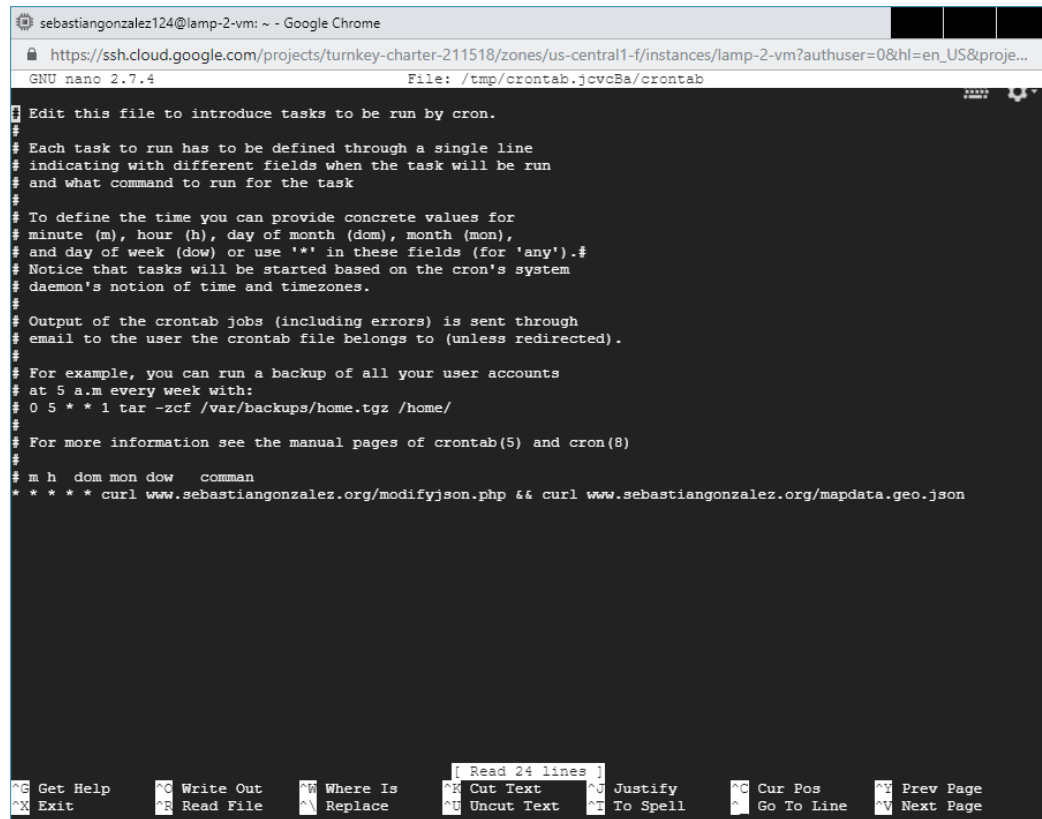
        )
    );
    # Add feature arrays to feature collection array
    array_push($geojson['features'], $feature);
}
$json = json_encode($geojson);
echo $json;
file_put_contents('mapdata.geo.json', $json);
?>

```

(Autores)

Figura 25. Código *modifyjson.php* parte2

Este script se ejecuta automáticamente cada minuto mediante una programación CRON, la cual es una herramienta de Linux que sirve para ejecutar tareas periódicamente.



```
sebastiangonzalez124@lamp-2-vm: ~ - Google Chrome
https://ssh.cloud.google.com/projects/turnkey-charter-211518/zones/us-central1-f/instances/lamp-2-vm?authuser=0&hl=en_US&proje...
GNU nano 2.7.4 File: /tmp/crontab.jcvcBa/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * curl www.sebastiangonzalez.org/modifyjson.php && curl www.sebastiangonzalez.org/mapdata.geo.json

^G Get Help      ^C Write Out    ^M Where Is     ^R Read 24 lines ^J Cut Text     ^O Justify      ^G Cur Pos     ^N Prev Page
^X Exit          ^S Read File    ^H Replace      ^U Uncut Text   ^T To Spell    ^_ Go To Line   ^V Next Page
```

(Autores)

Figura 26. CRON

Este código ejecuta a la página donde está ubicado *modifyjson.php* seguido de *mapdata.geo.json*.

El número de asteriscos determina el intervalo de tiempo entre cada ejecución de los scripts, para este proyecto se tienen cinco (5), por lo tanto, la acción se ejecuta a cada minuto.

Código *gotojs.js*:

```
setTimeout(function(){location = self.location.href;},60000);
//$( document ).ready(function() {
    var usuarioActual={
        user:"",
    };

    $.get("loginphp.php","",function(resp,status){
        usuarioActual.user=resp;
        $("#ident").html(usuarioActual.user);
        //window.alert("dentro de get "+usuarioActual.user);

        $.post("datosphp.php",usuarioActual,function(resp,status){
            var datos =JSON.parse(resp);
            //window.alert("POST")
            //window.alert(datos.INDICE);
            $("#PRT").html(datos.PRT);
            $("#HUM").html(datos.HUM);
            $("#TEMP").html(datos.TEMP);
            $("#CO").html(datos.CO);
            $("#CO2").html(datos.CO2);
            $("#O3").html(datos.O3);
            $("#GPS").html(datos.GPS);
            var longitud = datos.GPS.substring(0, datos.GPS.indexOf(',')) ;
            var latitud = datos.GPS.substring(datos.GPS.indexOf(',')+1,datos.GPS.length);
            $("#lat").html(latitud);
            $("#lon").html(longitud);
            //window.alert("long:"+longitud+"lat:"+latitud);
            if(datos.INDICE>=5){
                $("#estado").attr("class","column is-one-third has-background-danger");
            }
        })
    })
}
```

(Autores)

Figura 27. Código *gotojs.js* parte 1

```

        else if(datos.INDICE<5 && datos.INDICE>=3){
            $("#estado").attr("class","column is-one-third has-background-warning");
        }
        else{
            $("#estado").attr("class","column is-one-third has-background-success");
        }
    });
});
//});

$("#botongotousuario").click(function(){
    window.location.href='paginausuario.html';
});
$("#botongotomapa").click(function(){
    window.location.href='paginamapa.html';
});
});

```

(Autores)

Figura 28. Código *gotojs.js* parte 2

Este código utiliza el programa *datosphp.php* que se encarga de importar directamente los datos de la nube (desde la base de datos MYSQL).

Código *datosphp.php*:

```

<?php
$usuario="";
if($_SERVER['REQUEST_METHOD'] == "POST") {
    $host = "localhost";
    $dbname = "trabajodegrado";
    $user = "root";
    $pass = "dEBek7hvEBpiMp";
    $conectado = false;
    try {
        $pdo = new PDO('mysql:dbname='.$dbname.'.host='.$host,$user,$pass);
        $conectado = true;
    } catch(PDOException $e) {
        $conectado = false;
    }
    $usuario = htmlspecialchars($_POST["user"]);
    $query = $pdo->prepare("SELECT GPS, TIEMPO, O3, PRT, CO, CO2, TEMP, HUM, INDICE FROM datos WHERE usuario=\"".$usuario.\" ORD");
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    $json = json_encode($resultado[0]);
    echo $json;
}
?>

```

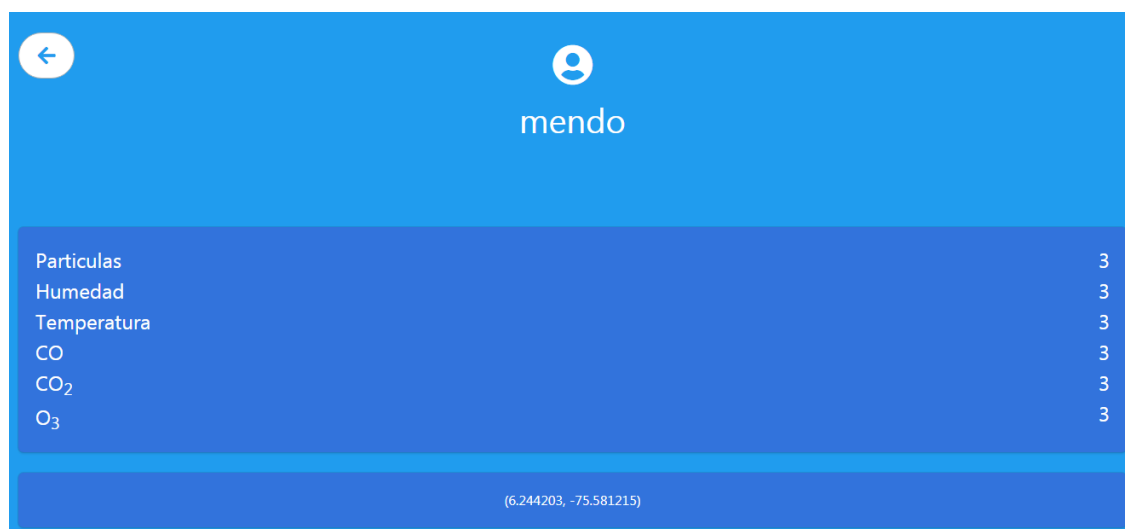
(Autores)

Figura 29. Código *datosphp.php*

Datosphp.php cuenta con toda la información necesaria para acceder a la base de datos (host, nombre de la base de datos, usuario y contraseña), dependiendo del usuario que ingresó, se escogen los datos asociados a esa cuenta y son los que se guardan en resultado para después ser enviados a las demás páginas.

Además, *paginamapa.html* cuenta con un botón para acceder a más información que se podrá observar en la *paginausuario.html*.

Información Sensores: Finalmente se tiene una interfaz en donde el usuario podrá ver los valores que está arrojando cada sensor para tener mayor información sobre cada variable que está participando en el cálculo. Para el estado de la calidad del aire, las variables son: ozono, dióxido de carbono, monóxido de carbono, partículas, temperatura y humedad. Además, se imprimen las coordenadas geográficas en donde se encuentra el dispositivo.



(Autores)

Figura 30. Información Sensores

Para la estructura de esta página, se tiene el siguiente el siguiente código HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.7.1/css/bulma.min.css">
  <link rel="shortcut icon" href="images/cloudicon.ico">
  <title>Cicclair</title>
  <script
src="https://code.jquery.com/jquery-3.2.1.js"
integrity="sha256-DZAnKJ/6XZ9si04HgrsXu/8s717jcIzLy3oi35EouyE="
crossorigin="anonymous">
  </script>
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.2.0/css/all.css" integrity="sha384-hwVjfwL6sN
</head>

<body class="has-background-info is-bold is-mobile">
  <section class="section">
    <div class="columns is-mobile">
      <div class="column is-one-fifth">
        <p class="is-size-1 has-text-white has-text-left">
          <a id="botongotomapa" class="button is-large is-rounded">
            <span class="icon has-text-info">
              <i class="fas fa-arrow-left"></i>
            </span>
          </a>
        </p>
      </div>
      <div class="column is-three-fifths">
        <p class="is-size-1 has-text-white has-text-centered">
```

(Autores)

Figura 31. Código *paginausuario.html* parte 1

```

        <a id="botongotomapa" class="button is-large is-rounded">
            <span class="icon has-text-info">
                <i class="fas fa-arrow-left"></i>
            </span>
        </a>
    </p>
</div>
<div class="column is-three-fifths">
    <p class="is-size-1 has-text-white has-text-centered">
        <i class="fas fa-user-circle"></i>
    </p>
    <p id="ident" class="is-size-2 has-text-white has-text-centered"></p>
</div>
</div>
</section>
<section class="section">
    <div class="box has-background-link">
        <div class="columns is-mobile">
            <div class="column has-text-left has-text-white is-size-4">
                <p>Particulas</p>
                <p>Humedad</p>
                <p>Temperatura</p>
                <p>CO</p>
                <p>CO<sub>2</sub></p>
                <p>O<sub>3</sub></p>
            </div>
            <div class="column has-text-right has-text-white is-size-4">
                <p id="PRT"></p>
                <p id="HUM"></p>
                <p id="TEMP"></p>
                <p id="CO"></p>
                <p id="CO2"></p>
                <p id="O3"></p>
            </div>
        </div>
    </div>
</section>
</div>
</div>
<div class="box has-background-link has-text-centered has-text-white">
    <p id="GPS">4 -72</p>
</div>
</section>
<script src="gotojs.js"></script>
</body>
</html>

```

(Autores)

Figura 32. Código *paginausuario.html* parte 2

```

    </div>
</div>
</div>
<div class="box has-background-link has-text-centered has-text-white">
    <p id="GPS">4 -72</p>
</div>
</section>
<script src="gotojs.js"></script>
</body>
</html>

```

(Autores)

Figura 33. Código *paginausuario.html* parte 3

Este HTML llama a un JavaScript llamado *gotojs.js* (explicado anteriormente) para toda la manipulación de las variables que están a la vista para ese usuario.

Se muestra cada variable junto con su valor en ese instante de tiempo. Actualmente hay un periodo de muestreo de 10 segundos para actualizar y mostrar los nuevos datos.

Finalmente, está el código encargado de recibir los datos del Arduino, llamado *ingresodatos.php*, el código es el siguiente:

```
<?php
$usuario="";
$GPS="";
$O3="";
$PRT="";
$CO="";
$CO2="";
$HUM="";
$TEMP="";
$INDICE="";
if($_SERVER['REQUEST_METHOD'] == "POST") {
    $host = "localhost";
    $dbname = "trabajodegrado";
    $user = "root";
    $pass = "dEBek7hvEBpiMp";
    $conectado = false;
    try {
        $pdo = new PDO('mysql:dbname='.$dbname.';host='.$host,$user,$pass);
        $conectado = true;
    } catch(PDOException $e) {
        $conectado = false;
    }
    $usuario = htmlspecialchars($_POST["user"]);
    $GPS = htmlspecialchars($_POST["gps"]);
    $O3 = htmlspecialchars($_POST["o3"]);
    $PRT = htmlspecialchars($_POST["prt"]);
    $CO = htmlspecialchars($_POST["co"]);
    $CO2 = htmlspecialchars($_POST["co2"]);
    $TEMP = htmlspecialchars($_POST["temp"]);
    $HUM = htmlspecialchars($_POST["hum"]);
    $INDICE = (((int)$O3)/10) + (((int)$PRT)/1000) + (((int)$CO)/10) + (((int)$CO2)/10);
    $query = $pdo->prepare("INSERT INTO `datos` (`usuario`, `GPS`, `TIEMPO`, `O3`, `PRT`, `CO`, `CO2`, `TEMP`, `HUM`, `INDICE`)
```

(Autores)

Figura 34. Código *ingresodatos.php* parte 1

```
    $query->execute();
    $resultado = $query->fetchAll(PDO::FETCH_ASSOC);
    $json = json_encode($resultado[0]);
    echo $json;
}
?>
```

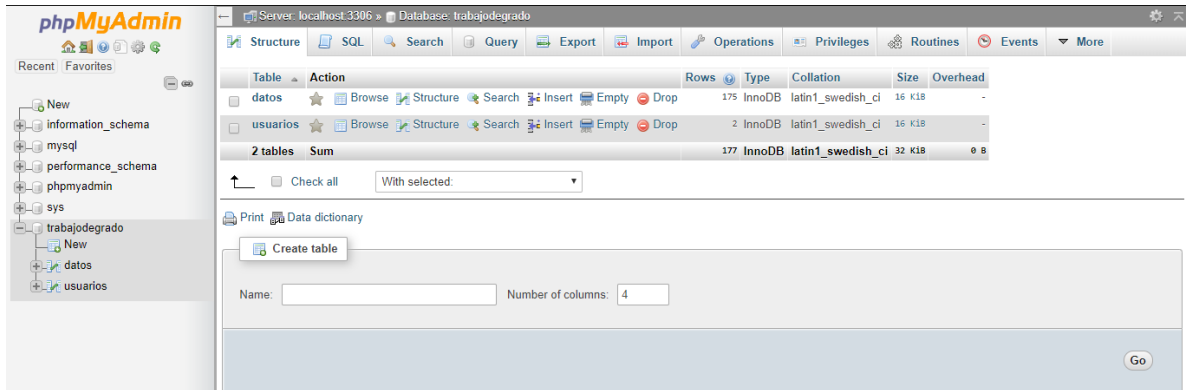
(Autores)

Figura 35. Código *ingresodatos.php* parte 2

Este recibe los datos de los sensores desde el dispositivo y en base a ellos calcula el índice para posteriormente almacenarlo en la base de datos. Cualquier modificación o calibración a los pesos de los indicadores debe hacerse modificando este código.

- **Base de Datos**

Como se mencionó previamente, se cuenta con una base de datos MySQL gestionada mediante phpMyAdmin, está compuesta por 2 tablas que se describen a continuación:

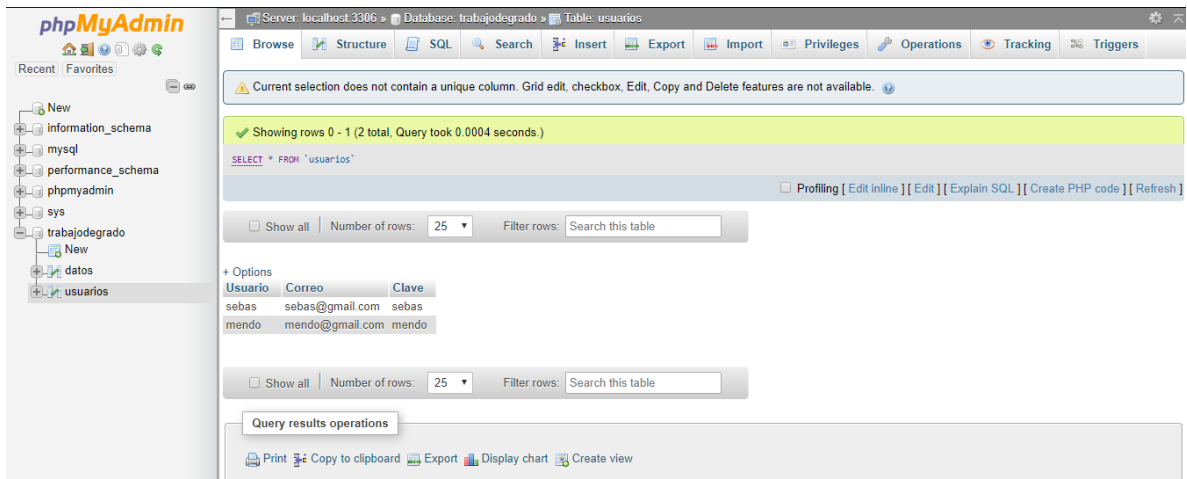


(Autores)

Figura 36. Interfaz de phpMyAdmin

Tabla Usuarios

Contiene la información de los usuarios registrados, es decir el nombre de usuario, correo electrónico y contraseña.

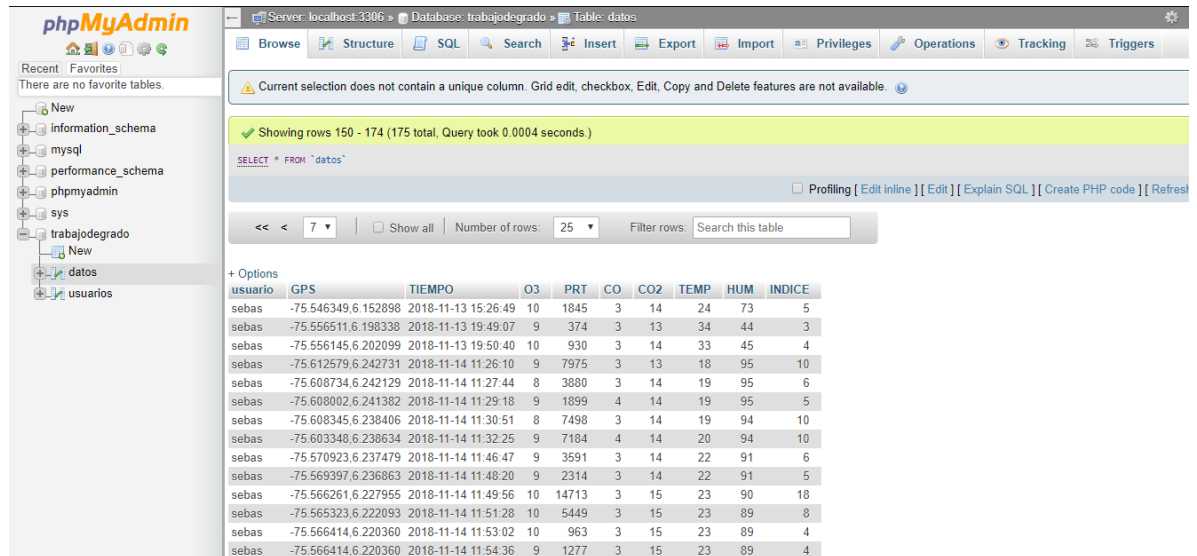


(Autores)

Figura 37. Tabla Usuarios

Tabla Datos

Contiene toda la información recopilada por los sensores del dispositivo, las cuales son el usuario, las coordenadas, el tiempo, y las variables de los sensores.

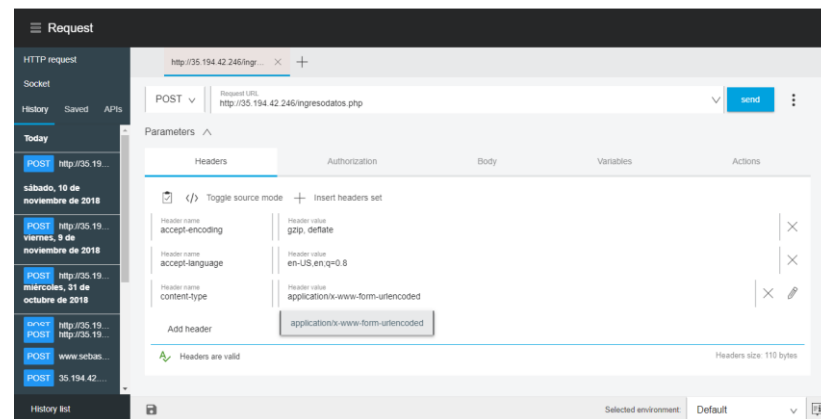


usuario	GPS	TIEMPO	O3	PRT	CO	CO2	TEMP	HUM	INDICE
sebas	-75.546349,6.152898	2018-11-13 15:26:49	10	1845	3	14	24	73	5
sebas	-75.556511,6.198338	2018-11-13 19:49:07	9	374	3	13	34	44	3
sebas	-75.556145,6.202099	2018-11-13 19:50:40	10	930	3	14	33	45	4
sebas	-75.612579,6.242731	2018-11-14 11:26:10	9	7975	3	13	18	95	10
sebas	-75.608734,6.242129	2018-11-14 11:27:44	8	3880	3	14	19	95	6
sebas	-75.608002,6.241382	2018-11-14 11:29:18	9	1899	4	14	19	95	5
sebas	-75.608345,6.238406	2018-11-14 11:30:51	8	7498	3	14	19	94	10
sebas	-75.603348,6.238634	2018-11-14 11:32:25	9	7184	4	14	20	94	10
sebas	-75.570923,6.237479	2018-11-14 11:46:47	9	3591	3	14	22	91	6
sebas	-75.569397,6.236863	2018-11-14 11:48:20	9	2314	3	14	22	91	5
sebas	-75.566261,6.227955	2018-11-14 11:49:56	10	14713	3	15	23	90	18
sebas	-75.565323,6.222093	2018-11-14 11:51:28	10	5449	3	15	23	89	8
sebas	-75.566414,6.220360	2018-11-14 11:53:02	10	963	3	15	23	89	4
sebas	-75.566414,6.220360	2018-11-14 11:54:36	9	1277	3	15	23	89	4

(Autores)

Figura 38. Tabla Datos

Se utiliza la herramienta Advanced REST Client para múltiples pruebas, sirve para crear peticiones http y así emular el envío de datos del dispositivo a la base de datos.



(Autores)

Figura 39. Advanced REST Client

3.4.2 Diseño Electrónico

El circuito electrónico se encuentra en los anexos 1, 2, 3, 4, 5.

Teniendo en cuenta el elevado precio del sensor de compuestos volátiles orgánicos y la filosofía de que el dispositivo sea asequible para cualquier público; se decidió quitar este sensor del dispositivo. Tras evaluar su relación costo/beneficio se concluye que el valor agregado que éste le genera al proyecto no es tan relevante. En su reemplazo se tiene el sensor de partículas para el cálculo de la calidad del aire.

- **Acondicionamiento de señales**

Módulo AM2302 (Temperatura, Humedad): No necesita acondicionamiento de la señal, se utiliza la librería SimpleDHT, Esta librería extrae la temperatura en grados Celsius y la Humedad en porcentaje de humedad relativa directamente.

PPD42NS: Este sensor trabaja con 3 variables, cantidad de tiempo de pulso en bajo (Mientras una partícula corta o tapa un haz de luz infrarroja) en un intervalo de tiempo (en este caso 30s), el ratio es el porcentaje del tiempo en pulso bajo y finalmente de la gráfica del sensor se tiene que: $Concentracion = 1.1 * ratio^3 - 3.8 * ratio^2 + 520 * ratio + 0.62$

Físicamente, el circuito no necesita ningún tipo de acondicionamiento.

Módulo MQ-131 (Ozono), Módulo MQ-7 (Monóxido de Carbono), MG-811 (Dioxido de Carbono): Estos sensores no tienen circuito de acondicionamiento, se esta trabajando con porcentajes. El puerto analogo lee el valor que arrojan los sensores, los divide por 1023 y los convierte en porcentajes para proceder a las comparaciones. Estos tampoco necesitan un circuito de acondicionamiento.

GPRS/GPS (AI-Thinker A7): Es un módulo con funciones de GPS GPRS GSM. Funciona con comandos AT, puede enviar datos por HTTP, TCP/IP y cuenta con localización por satélite usando el GPS. Viene con 2 antenas (GSM y GPS), la antena GPS viene con una frecuencia de 2.4GHz (comúnmente usado en wifi), está antena funciona muy bien en zonas con perfectas condiciones, es decir al aire libre, Se vio la necesidad de reemplazar esta antena por otra con una frecuencia de 1575.42 MHz ya que ésta es la frecuencia adecuada para trabajar con los satélites. El módulo tiene un puerto serial para comunicación dedicado que funciona a 115200 bauds (por donde se enviaran los comandos AT). Además, tiene otro canal adicional que funciona a 9600 bauds por donde solo mandará la información del GPS con la estructura anunciada en la norma NMEA.

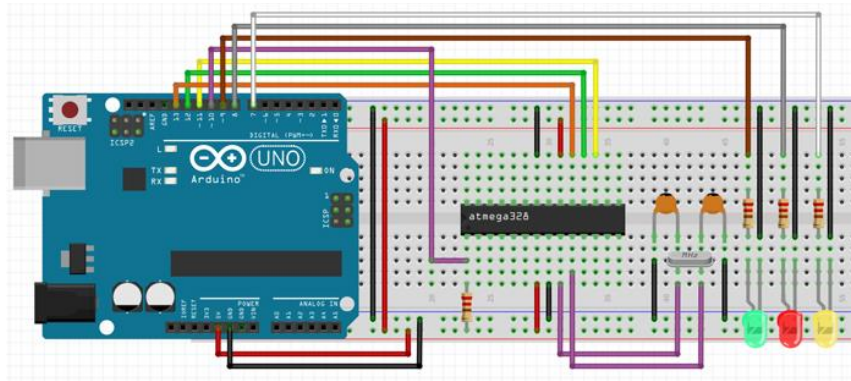
Para la lectura de la información del AI – Thinker A7 se utiliza una librería que se llama *TinyGPS++*, Ésta librería Interpreta los datos en formato NMEA y genera la información pertinente, es decir, las coordenadas actuales que está leyendo el GPS.

- **Procesamiento de los datos**

Todos los datos generados por los sensores mencionado anteriormente se almacenan en el microcontrolador Atmega 328P, Toda la programación de este microcontrolador se realizó desde el software Arduino y cuando el código estuvo listo y funcional, se programó en el Atmega 328P utilizando Boot Loader.

Boot Loader: Es un conjunto de instrucciones que nos permiten acceder al microcontrolador sin tener que conocer instrucciones muy avanzadas.

Utilizando el Arduino como programador ISP con la siguiente conexión como se puede observar en la figura 6, se pasa el programa al Atmega 328P



(Minitronica, 2016)

Figura 40. Conexión Arduino – Atmega

El código que se utiliza para este trabajo de grado tiene 3 funciones:

- **Lectura de variables**

Utiliza los puertos de entrada análogos o digitales, para recibir los datos enviados por los sensores. En los casos del sensor de partículas y del GPRS/GPS se hace uso de las diferentes librerías antes de almacenar dicho dato en una variable.

- **Procesamiento de los datos**

El programa hace las operaciones necesarias para dejar todas las variables en términos que se puedan comparar, para este caso se está utilizando porcentajes.

- **Enviar la información a la Base de datos**

Se envía la información a una base de datos. Para la conexión con el servidor se utilizan los comandos AT en el AI-THINKER A7:

Los comandos AT son mensajes que se envían por el puerto serial al AT-Thinker (forma de comunicarse con el modulo) junto con una instrucción. Las instrucciones utilizadas en este proyecto fueron las siguientes:

AT+CIPSTART=TCP,www.sebastiangonzalez.org,80

Este comando sirve para indicar que con el protocolo TCP se iniciará una conexión con la dirección www.sebastiangonzalez.org por el canal 80.

AT+CIPSEND

Este comando inicia la comunicación con el servidor.

Para el envío de estos datos se utilizan peticiones HTTP.

El método a utilizar es post: El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

Finalmente, después de que se envía toda la información, se cierra la conexión con: AT+CIPCLOSE

A continuación, se puede observar el código en arduino el cual está debidamente documentado mediante comentarios:

```
#include <SoftwareSerial.h> //se necesita un puerto serial dedicado a la señal gps por tal motivo se emplea la libreria

#include <SimpleDHT.h> //libreria para usar modulo am2302
#include <TinyGPS++.h> //libreria para interpretar GPS

//declaracion de pines de los sensores
#define pin_co A1 //requieren pin analogo ya que el modulo lo entrega
#define pin_co2 A2
#define pin_o3 A3
#define pin_particulas 7 //el modulo entrega señal digital
#define pin_humtemp 8

String user="sebas"; //identificacion de usuario requerido por el servidor

SoftwareSerial gpsSerial(13,3); //puerto serial para señal gps RX,TX
SimpleDHT22 dht; //declaracion de objeto dht tipo SimpleDHT22 de libreria
TinyGPSPlus gps; //declaracion de objeto gps de libreria

unsigned long tiempo_muestreo = 90000; //30s tiempo de muestreo
unsigned long tiempo_pasado; //tiempo de ultima medicion

//declaracion e iniciacion variables ppd42ns
unsigned long duration;
unsigned long lowpulseoccupancy = 0;
float ratio = 0;
(Autores)
```

Figura 41. Código en Arduino parte 1

```

//declaracion e iniciacion variables de medicion
float particulas = 0; //particulas/283mL
float humidity = 0; %%
float temperature = 0; //°C
int co = 0; %%
int co2 = 0; %%
int o3 = 0; %%
double longitude=0.0;
double latitude=0.0;

void setup() {
  Serial.begin(115200); //inicia comunicacion baud debe ser 115200 por el gprs
  gpsSerial.begin(9600); //inicia comunicacion baud debe ser 9600 por el pin gps del A7

  //decalaracion de pines como entradas
  pinMode(pin_co,INPUT);
  pinMode(pin_co2,INPUT);
  pinMode(pin_o3,INPUT);
  pinMode(pin_particulas,INPUT);

  tiempo_pasado = millis();//tiempo inicio
  gpsSerial.println("Esperando inicializacion de GPS 20s");
  delay(20000); //espera mientras el modulo se inicializa
  Serial.println("AT+GPS=1"); //empieza a rastrear el GPS y enviar la informacion por el puerto dedicado
  esperarOK();
  gpsSerial.println("GPS OK 90s");
  delay(90000); //el modulo se demora en encontrar los satelites
  gpsSerial.println("Satelites conectados");
}

```

(Autores)

Figura 42. Código en Arduino parte 2

```

}

void loop() {
  //coordenadas GPS
  while (gpsSerial.available() > 0)
    gps.encode(gpsSerial.read());
  if (gps.location.isUpdated()){
    longitude=gps.location.lng();
    latitude=gps.location.lat();
  }

  duration = pulseIn(pin_particulas, LOW); //detecta pulso bajo del sensor de particulas
  lowpulseoccupancy = lowpulseoccupancy+duration; //acumula pulso bajo

  if ((millis()-tiempo_pasado) >= tiempo_muestreo) { //lleva 30s acumulando
    ratio = lowpulseoccupancy/(tiempo_muestreo*10.0); //porcentaje del tiempo en pulso bajo
    particulas = 1.1*pow(ratio,3)-3.8*pow(ratio,2)+520*ratio+0.62; //curva caracteristica convierte porcentaje de
    //tiempo en pulso bajo a particuals/283 mL

    dht.read2(pin_humtemp,&temperature,&humidity, NULL); //lee los valores de humedad y temperatura del sensor dht
    // y los asigna a las variables

    //lee valores analogos de los sensores y los mapea a porcentaje
    co = analogRead(pin_co);
    co = map(co, 0, 1023, 0, 100);
    co2 = analogRead(pin_co2);
    co2 = map(co2, 0, 1023, 0, 100);
    o3 = analogRead(pin_o3);
    o3 = map(o3, 0, 1023, 100, 0);
  }
}

```

(Autores)

Figura 43. Código en Arduino parte 3

```

    if (longitud+latitud!=0){
        comm(particulas,humidity,temperature,co,co2,o3,latitud,longitud); //envia los datos
    }
    lowpulseoccupancy = 0; //reinicia acumulador
    tiempo_pasado = millis(); //reinicia contador
}
}

void comm(float comm_particulas, float comm_humidity, float comm_temperature, int comm_co, int comm_co2, int comm_o3,
double comm_longitude, double comm_latitude){
    //crea la cadena que sera enviado en el POST y determina su longitud
    String datosconcatenados = "user="+user+"&gps="+String(comm_latitude,6)+" "+String(comm_longitude,6)+"&o3="
+String(comm_o3)+"&prt="+String(comm_particulas)+"&co="+String(comm_co)+"&co2="+String(comm_co2)+"&hum="
+String(comm_humidity)+"&temp="+String(comm_temperature);

    String datolength = String(datosconcatenados.length());
    gpsSerial.println("Datos a enviar:" + datosconcatenados);
    Serial.println("AT+CIPSTART=\"TCP\", \"35.194.42.246\",80"); //se conecta al servidor
    esperarOK();
    gpsSerial.println("Conectado al servidor");
    delay(500);
    Serial.println("AT+CIPSEND");//empieza comunicacion con el servidor
    delay(1000);
    //encabezado de la peticion HTTP
    Serial.print("POST /ingresodatos.php HTTP/1.1\r\n");
    Serial.print("Host: www.sebastiangonzalez.org\r\n");
    Serial.print("accept-encoding: gzip, deflate\r\n");
    Serial.print("accept-language: en-US,en;q=0.8\r\n");
}
(Autores)

```

Figura 44. Código en Arduino parte 4

```

Serial.print("content-type: application/x-www-form-urlencoded\r\n");
Serial.print("Content-Length: "+ datolength+"\r\n\r\n");
//cuerpo de la peticion HTTP
Serial.println(datosconcatenados);
delay(500);
Serial.println('\x1A'); //ctrl+z para marcar fin de comunicacion
delay(500);
Serial.println("AT+CIPCLOSE"); //cierra coneccion con el servidor
gpsSerial.println("Datos enviados");
}

void esperarOK(){ //espera hasta que detecta "OK"
    while(Serial.read() != 'K') {}
}
(Autores)

```

Figura 45. Código en Arduino parte 5

El dispositivo cuenta con un puerto para realizar debugging conectado al Tx del serial “gpsSerial”. A continuación, se observa la información recibida de dicha interfaz:

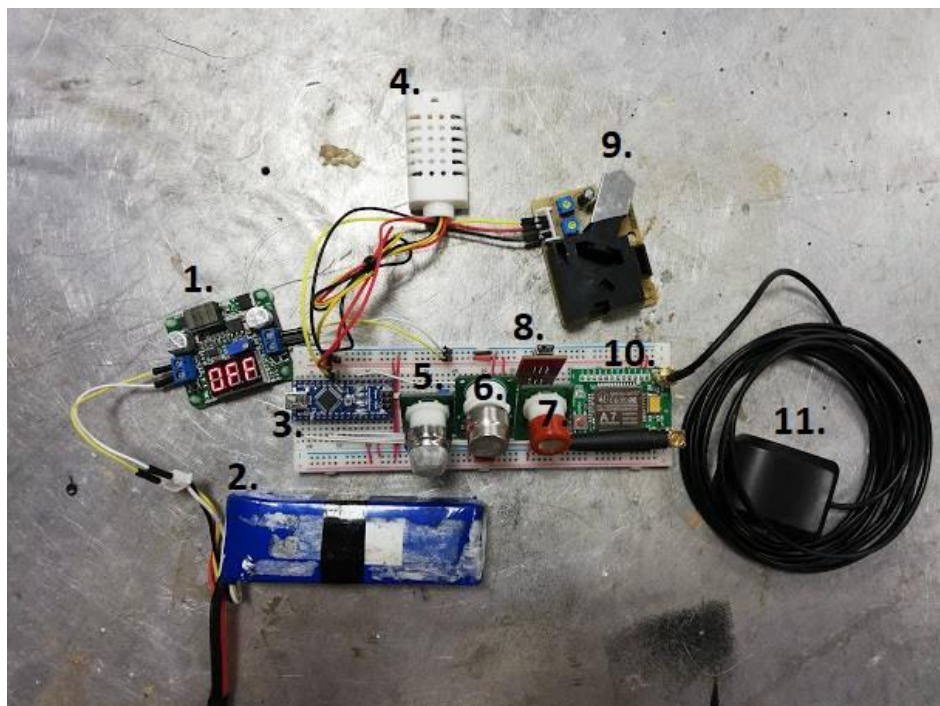
```
Esperando inicializacion de GPS 20s
GPS OK 90s
Satelites conectados
Datos a enviar:user=sebas&gps=-75.516983,6.156582&o3=7&prt=1862.66&co=0&co2=12&hum=45.60&temp=27.50
Conectado al servidor
Datos enviados
Datos a enviar:user=sebas&gps=-75.516975,6.156578&o3=8&prt=2862.06&co=0&co2=11&hum=46.50&temp=27.50
Conectado al servidor
Datos enviados
```

(Autores)

Figura 46. Respuesta por el puerto serial

De esta manera se verifica el funcionamiento del dispositivo.

El prototipo construido en protoboard para sus pruebas quedó de la siguiente forma:



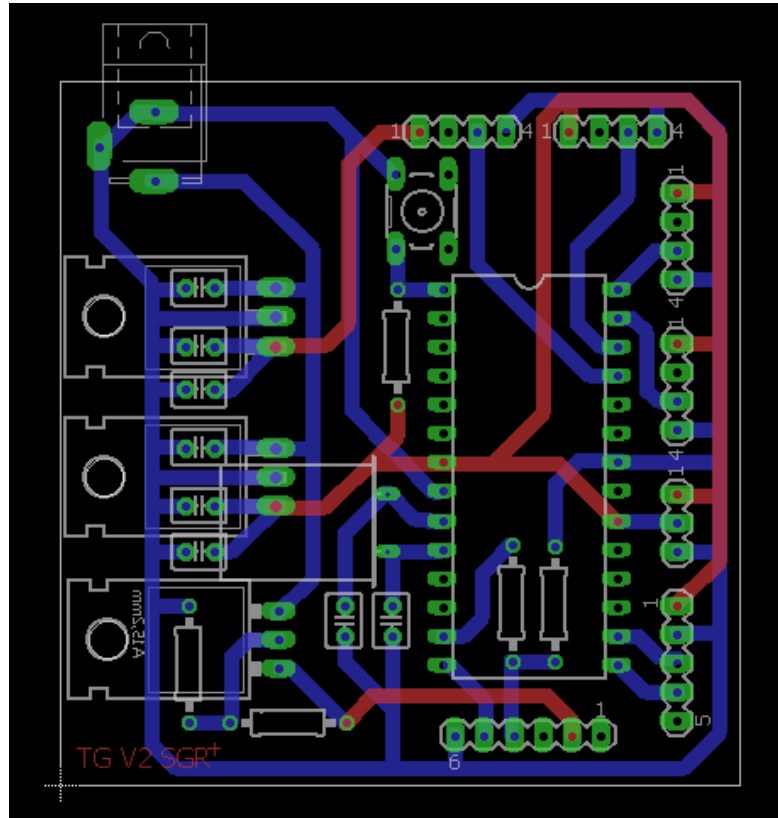
(Autores)

Figura 47. Circuito montado en protoboard

Componentes utilizados en el circuito (ver figura 47):

1. Regulador de voltaje
2. Batería recargable Li-Po 11v 2200mAh
3. Arduino nano
4. Módulo DHT22 (AM2302)
5. Módulo (ozono)
6. Módulo (Co2)
7. Módulo (Co)
8. Puerto Serial para Debug
9. PPD42NS
10. AI-Thinker A7
11. Antena GPS

Si bien el prototipo creado en protoboard es funcional, se buscó dar valor agregado al proyecto, diseñando una posible versión 2.0 del dispositivo en un circuito impreso el cual se encuentra en los anexos.

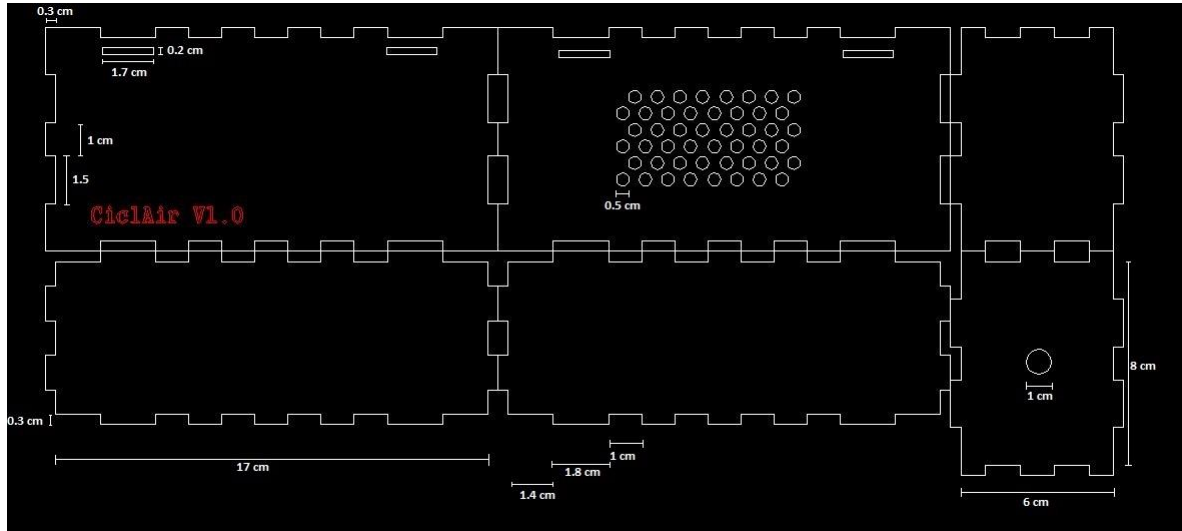


(Autores)

Figura 48. Circuito impreso

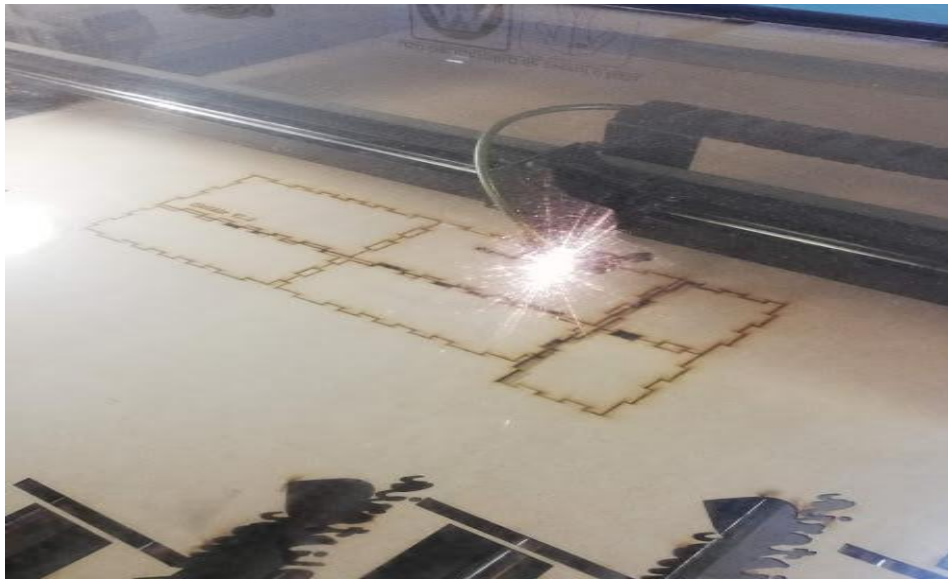
3.4.3 Diseño Mecánico

El prototipo tiene una estructura desarrollada a base de MDF, los cortes del material se hicieron con láser utilizando el siguiente diseño:



(Autores)

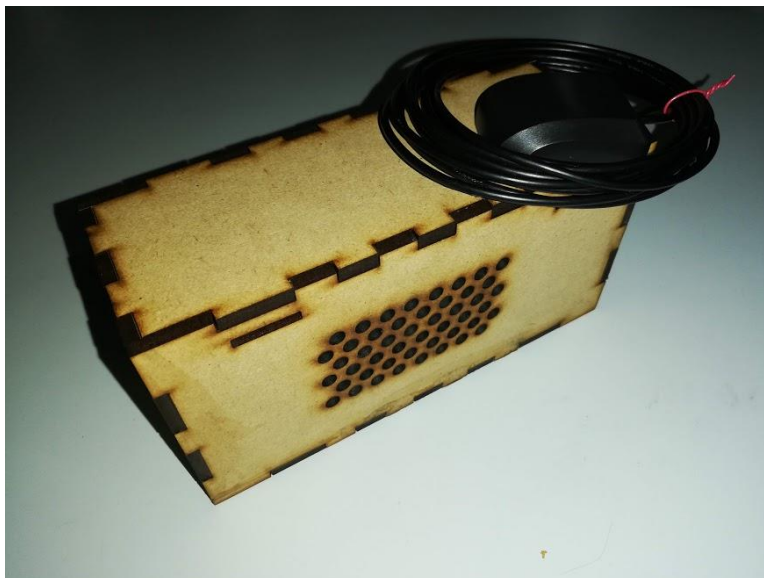
Figura 49. Diseño estructura dispositivo



(Autores)

Figura 50. Proceso corte láser

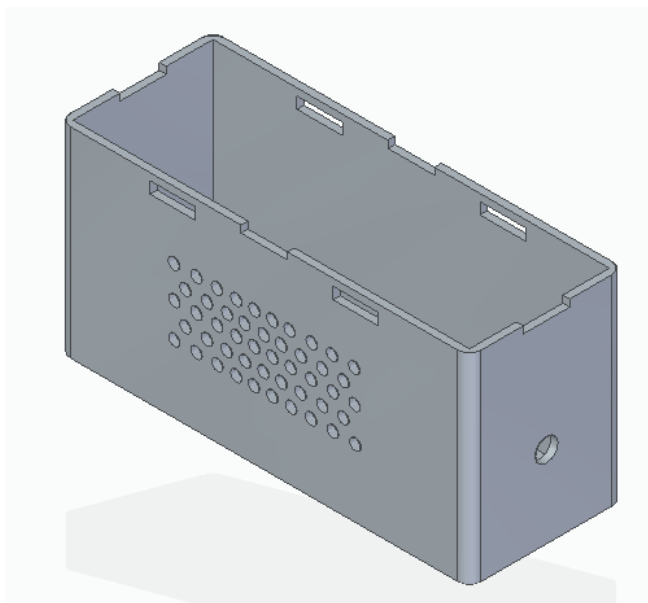
La caja está compuesta de 6 caras, cuenta con los huecos necesarios para el flujo de aire a los sensores, tiene inscrito con láser el nombre del dispositivo y ensamblada se ve de la siguiente forma:



(Autores)

Figura 51. Prototipo final

Adicionalmente se hizo el diseño de la misma estructura para una versión 2.0 del dispositivo pensada para imprimirse en una 3D dándole mayor robustez y seguridad al dispositivo.



(Autores)

Figura 52. Modelo 3d de la estructura

El dispositivo se debe acoplar a la bicicleta de tal forma que no se interponga en la operatividad de la bicicleta, por lo tanto, se decide ubicarlo en el tubo justo debajo del sillín de la bicicleta.



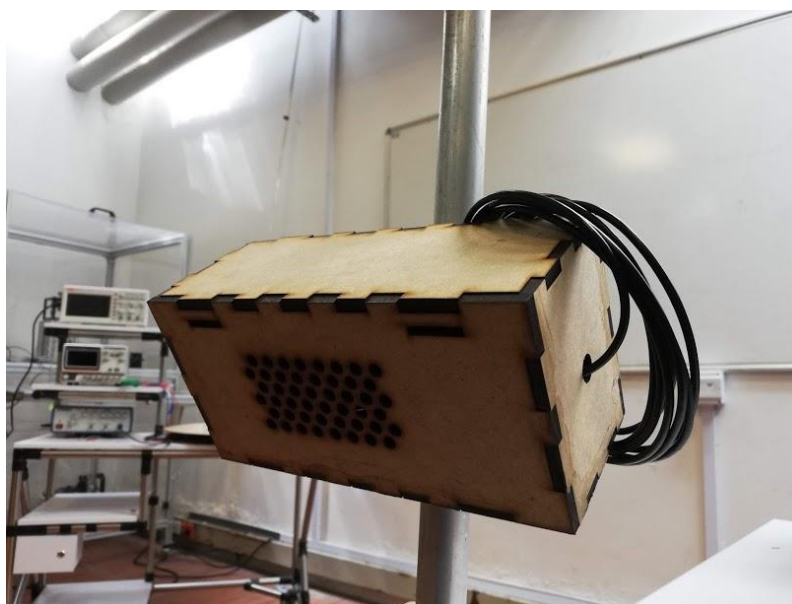
(Autores)

Figura 53. Prototipo acoplado vista 1



(Autores)

Figura 54. Prototipo acoplado vista 2



(Autores)

Figura 55. Prototipo acoplado vista 3

3.5 PRUEBAS Y REFINAMIENTO

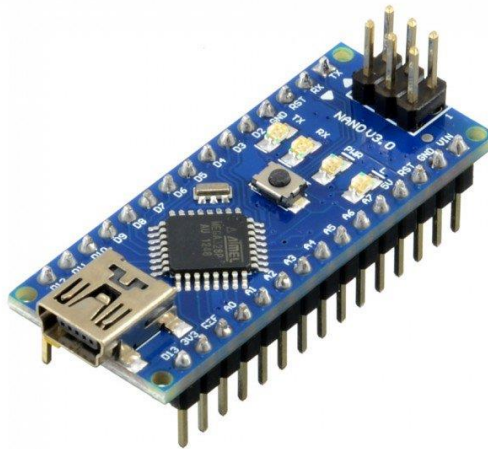
El funcionamiento del módulo AI-Thinker A7 se comprueba enviando manualmente los comandos AT desde el puerto serial de un computador.

Se utilizó ARC para simular la señal que envía el dispositivo y así garantizar el funcionamiento de la página web.

Se hizo un recorrido por la ciudad, pasando por puntos específicos de interés para verificar el éxito del envío de datos a la plataforma, utilizando el dispositivo sin calibrar.

Se verifica el funcionamiento del sistema completo, se hace una calibración básica. Para calibrar el dispositivo se utilizan datos de estaciones del Siata. Se modifican los pesos de las variables dentro del código *ingresodatos.php*, de tal forma que se genere un índice comparable.

Debido a inconvenientes con el proceso de Bootloading del Atmega328p, se tomó la decisión de migrar la programación a un Arduino Nano. Este dispositivo cuenta con una huella física similar a la del integrado. Por lo cual, no se ve muy afectado el diseño del sistema.



(Iberobotics, 2018)

Figura 56. Arduino Nano V 3.0

4. RESULTADOS

Acorde a lo estipulado en los objetivos, se lograron los siguientes resultados:

4.1 PROTOTIPO FUNCIONAL

Se logró construir un dispositivo funcional, portable, capaz de leer, acondicionar y enviar las variables estipuladas. Adicionalmente, sus dimensiones son acordes a lo delimitado en el alcance y es acoplable a una bicicleta. Su desempeño es validado en las pruebas realizadas.



(Autores)

Figura 57. Prototipo acoplado a una bicicleta



(Autores)

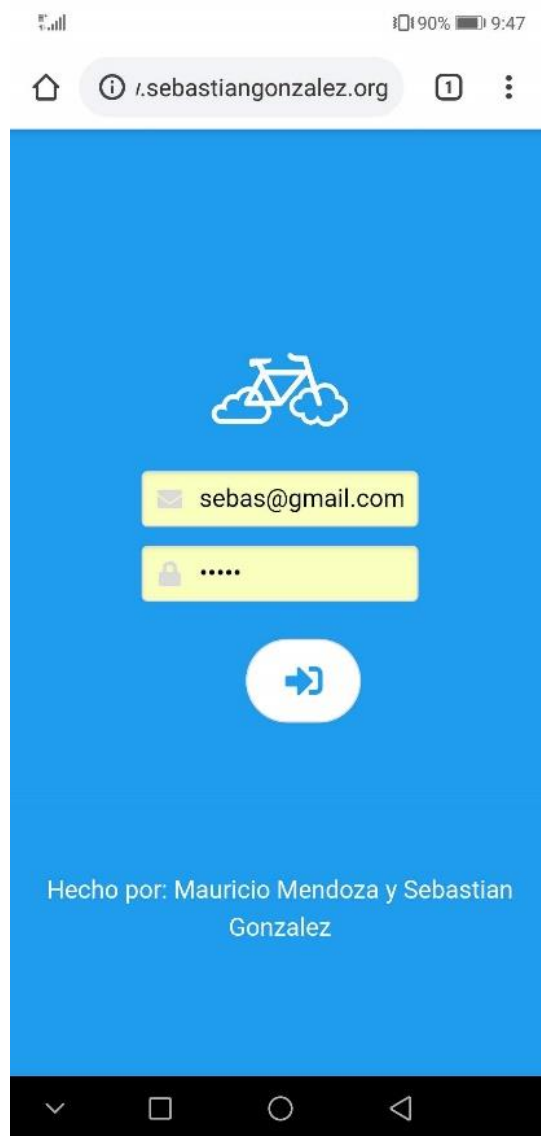
Figura 58. Prototipo acoplado a una bicicleta 2

Se puede ver en las figuras 57 y 58 que la ubicación y el tamaño del dispositivo no entorpecen la operatividad de la bicicleta.

4.2 PÁGINA WEB

Se desarrolló una página web funcional, hecha con diseños propios del equipo del proyecto, cuenta con una interfaz agradable al usuario y todo lo demás descrito anteriormente. Adicionalmente, se compró un dominio para esta: www.sebastiangonzalez.org.

- **Interfaz Inicial**



(Autores)

Figura 59. Interfaz inicial

- **Interfaz Mapa**

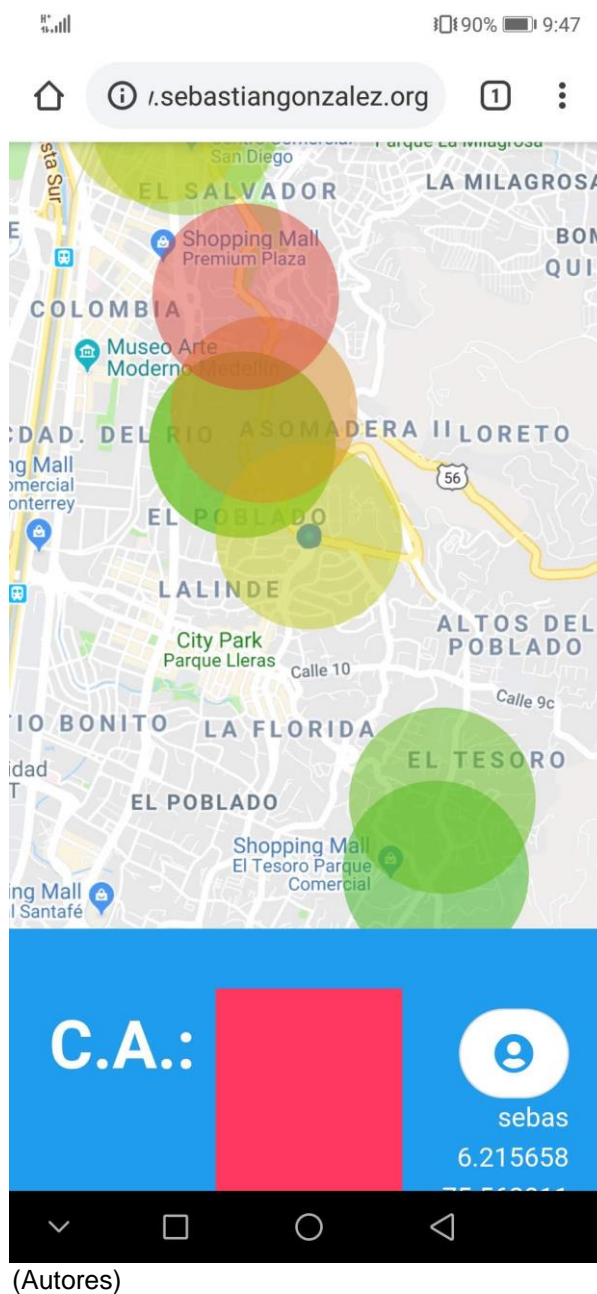


Figura 60. Interfaz mapa

- **Interfaz de Información Sensores**

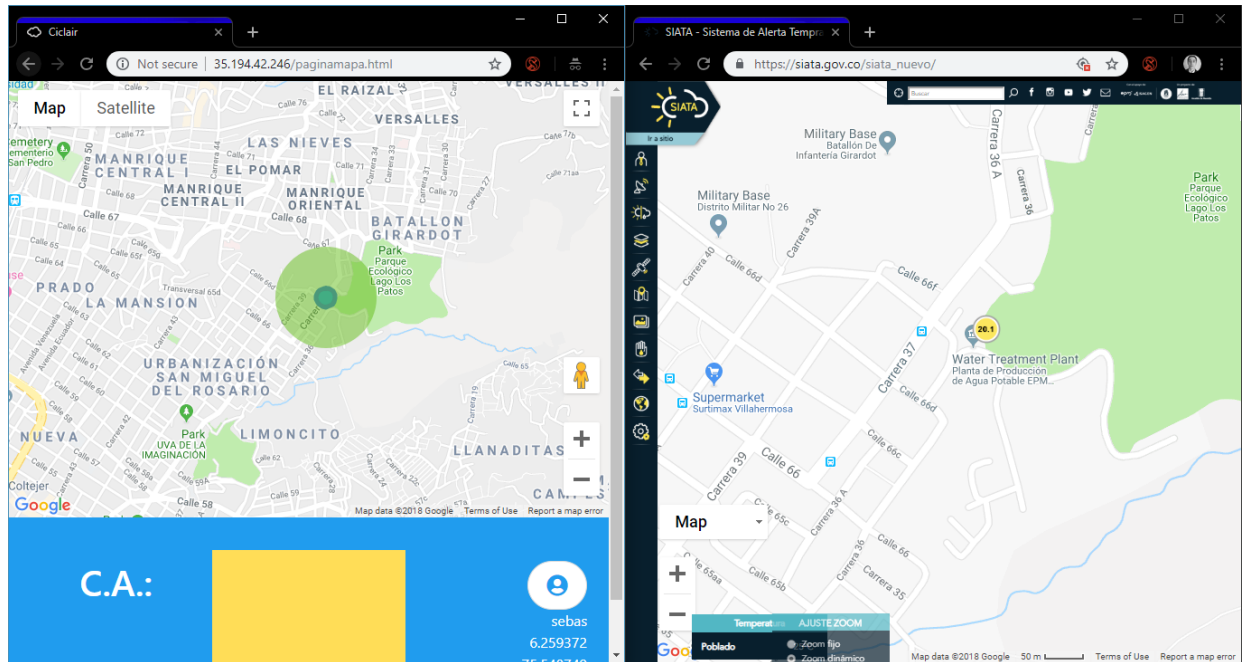


(Autores)

Figura 61. Interfaz Información Sensores

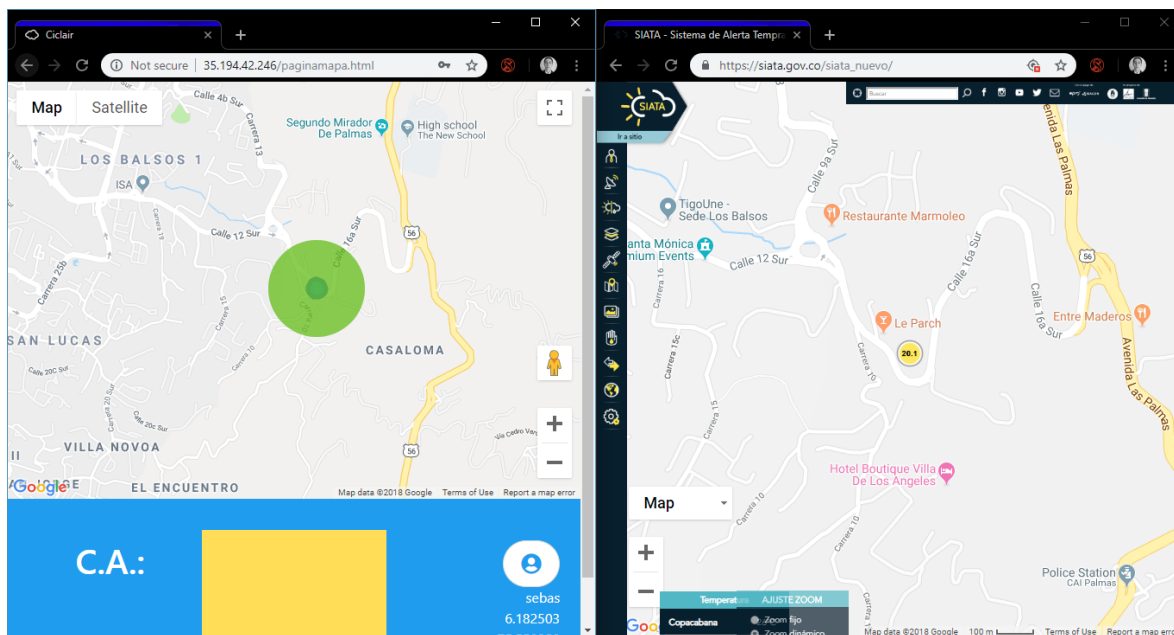
- **Validación de la información**

A continuación, se tiene una comparación de los datos tomados en ubicaciones cercanas a donde se encuentran los sensores actuales del Siata para hacer una validación de la información arrojada por el prototipo.



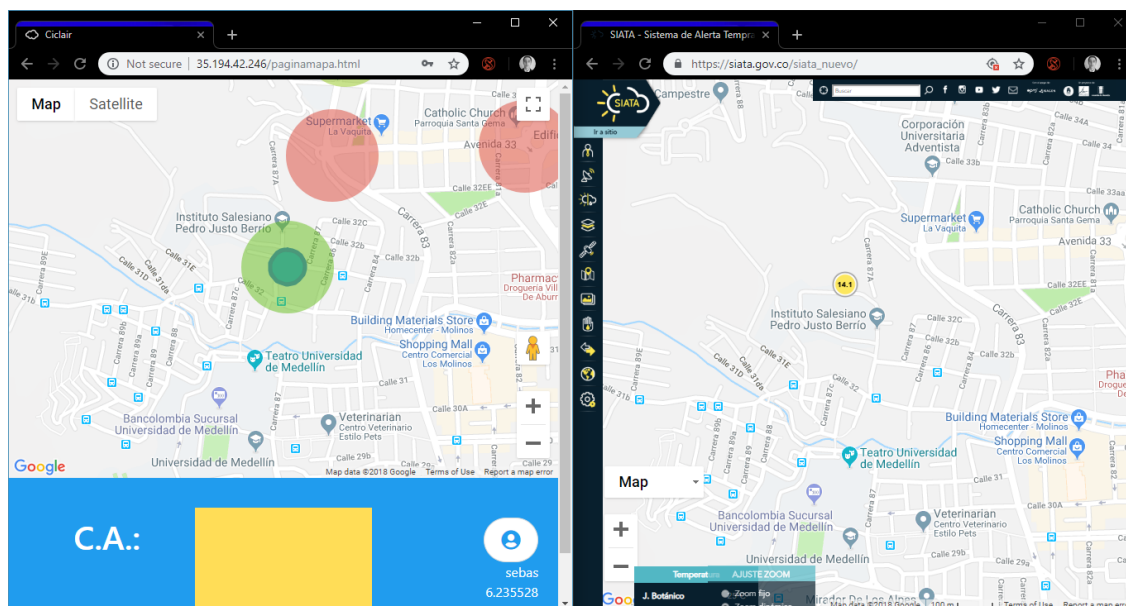
(Autores)

Figura 62. Comparación con el Siata por Parque Ecológico Lago Los Patos



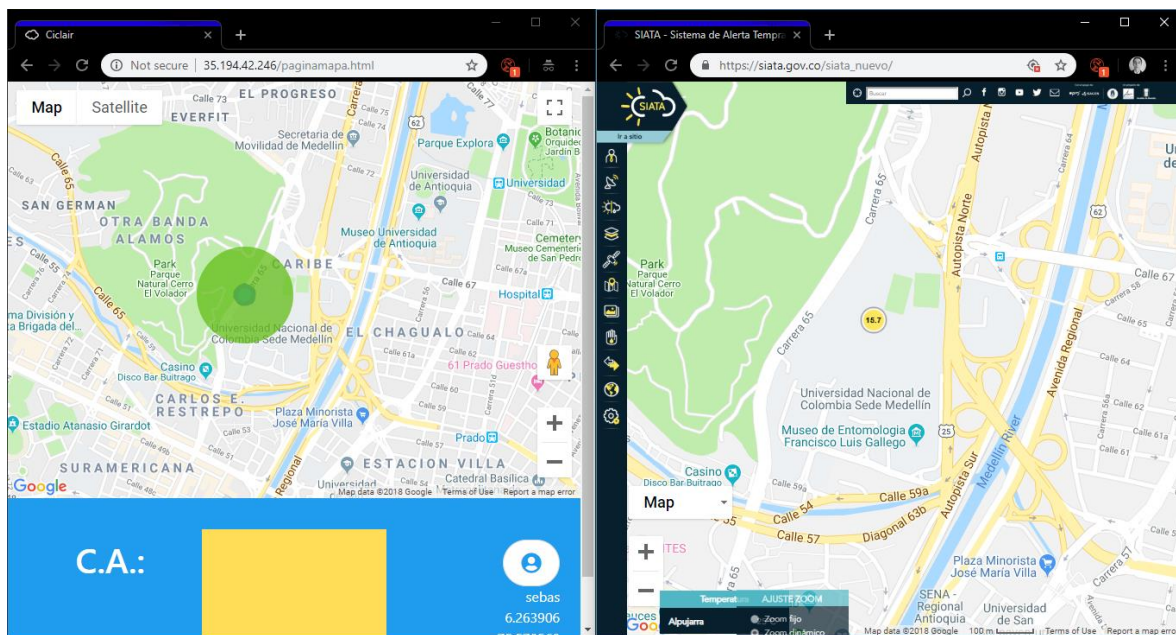
(Autores)

Figura 63. Comparación con el Siata por la vía Palmas



(Autores)

Figura 64. Comparación con el Siata por el Instituto Salesiano Pedro Justo Berrio



(Autores)

Figura 65. Comparación con el Siata en la Universidad Nacional de Colombia

Se puede observar en las imágenes 62, 63, 64 y 65 que los datos arrojados por el dispositivo coinciden en el estado en el que se encuentra la calidad del aire. Gracias a la calibración que se hizo en la sección de pruebas y refinamiento, se tiene una escala comparable a aquella utilizada por el Siata confirmando así la veracidad de los datos.

5. CONCLUSIONES Y CONSIDERACIONES FINALES

Es necesario hacer una debida calibración de los sensores para leer correctamente las variables físicas del entorno en donde se va a usar el dispositivo. Se debe tener en cuenta que algunos sensores pueden cambiar su precisión a medida que cambia la temperatura o la presión. Por lo anterior, es necesario realizar pruebas en distintos lugares con diversas condiciones para definir una calibración apropiada del dispositivo.

Inicialmente, los números que arrojaron los sensores no estaban en la misma escala del Siata (Sistema de Alerta Temprana de Medellín y el Valle de Aburrá). Para poder comparar los datos generados y validarlos fue necesario realizar un ajuste en el código *ingresodatos.php*. Así acoplándolos a una escala de colores similar.

En el mercado existen diversos productos que miden calidad del aire. Lo importante finalmente no son los datos en sí, si no la veracidad de estos. Es decir, generar la confianza en el cliente, en cuanto a que los datos son correctos. Esto se logra utilizando como ejemplo a alguna entidad de confianza (en este caso el Siata) para validarlos. Además, tener en cuenta lo que se hace con esa información generada. En este caso el sistema ofrece la posibilidad de visualizarlos en un mapa en tiempo real actualizado por múltiples usuarios. Aquellos detalles son los que le dan un valor agregado al dispositivo que se encarga de ponerlo a competir con los otros del mercado.

Para el ahorro de recursos en el desarrollo del proyecto es fundamental hacer un estudio inicial para determinar cuáles son las variables influenciales que dictaminan la calidad del aire.

En este trabajo de grado se pudo observar que el sensor de compuestos orgánicos volátiles no generaba el suficiente valor agregado al dispositivo a pesar de ser tan costoso, en su lugar se utilizara el sensor de partículas pm 2.5 que es mucho más económico y si le genera un gran valor al proyecto.

El contenido del presente trabajo de grado, junto con los códigos y demás recursos son de carácter libre. La idea es aportar los conocimientos adquiridos a la comunidad para que cuenten con una base para proyectos similares.

REFERENCIAS

- Aguilar, J. L., & Leiss, E. (Septiembre de 2004). *Introducción a la Computación Paralela*. Obtenido de [ing.ula.ve: http://www.ing.ula.ve/~aguilar/publicaciones/objetos/libros/ICP.pdf](http://www.ing.ula.ve/~aguilar/publicaciones/objetos/libros/ICP.pdf)
- Álvarez, V. A. (6 de Diciembre de 2015). *Por cada 3 habitantes hay un vehículo rodando en Medellín*. Obtenido de <http://www.elcolombiano.com: http://www.elcolombiano.com/antioquia/movilidad/en-medellin-transita-un-carro-por-cada-tres-habitantes-EB3232363>
- Andunce, F. (17 de Junio de 2014). *Alerta Ambiental: ¿Cómo afecta a los ciclistas la mala calidad del aire?* Obtenido de <https://www.guioteca.com: https://www.guioteca.com/ciclismo-urbano/alerta-ambiental-como-afecta-a-los-ciclistas-la-mala-calidad-del-aire/>
- Arias, X. (15 de 6 de 2017). *Aunque usted no lo crea, montar la bicicleta afecta su respiración*. Obtenido de [Publimetro: https://www.publimetro.co/co/bogota/2017/06/15/por-el-aire-que-respira-montar-en-bicicleta-puede-ser-un-riesgo-para-su-salud.html](https://www.publimetro.co/co/bogota/2017/06/15/por-el-aire-que-respira-montar-en-bicicleta-puede-ser-un-riesgo-para-su-salud.html)
- Belloch, C. (2006). *LAS TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN (T.I.C.)*. Obtenido de [uv.es: https://www.uv.es/~bellochc/pdf/pwtic1.pdf](https://www.uv.es/~bellochc/pdf/pwtic1.pdf)
- Benito, F. (6 de 12 de 2015). *Protege tu salud midiendo la contaminación de tu alrededor con este dispositivo*. Obtenido de [Omicronno: https://omicronno.elespanol.com/2015/12/atmotube-el-wearable-para-medir-la-contaminacion-por-terminar/](https://omicronno.elespanol.com/2015/12/atmotube-el-wearable-para-medir-la-contaminacion-por-terminar/)
- Botero, J. D. (2 de 1 de 2017). *Este es flow, El aparato que mide la calidad del aire que respiras*. Obtenido de <http://www.enter.co/eventos/ces/2017/conoce-a-flow-el-aparato-que-mide-la-calidad-del-aire-que-respiras/>
- Correa, M. V. (4 de Noviembre de 2016). *Antioquia es el departamento con más emisiones de CO2 en el país*. Obtenido de <http://www.elcolombiano.com: http://www.elcolombiano.com/colombia/estos-son-los-departamentos-con-mas-emision-de-gases-de-efecto-invernadero-FA5299887>
- De la Cruz, L. A. (8 de Octubre de 2015). *Qué es la electrónica, para qué nos sirve y cómo la podemos utilizar?* Obtenido de ingenieriaelectronica.org: https://ingenieriaelectronica.org/que-es-la-electronica-para-que-nos-sirve-y-como-la-podemos-utilizar/

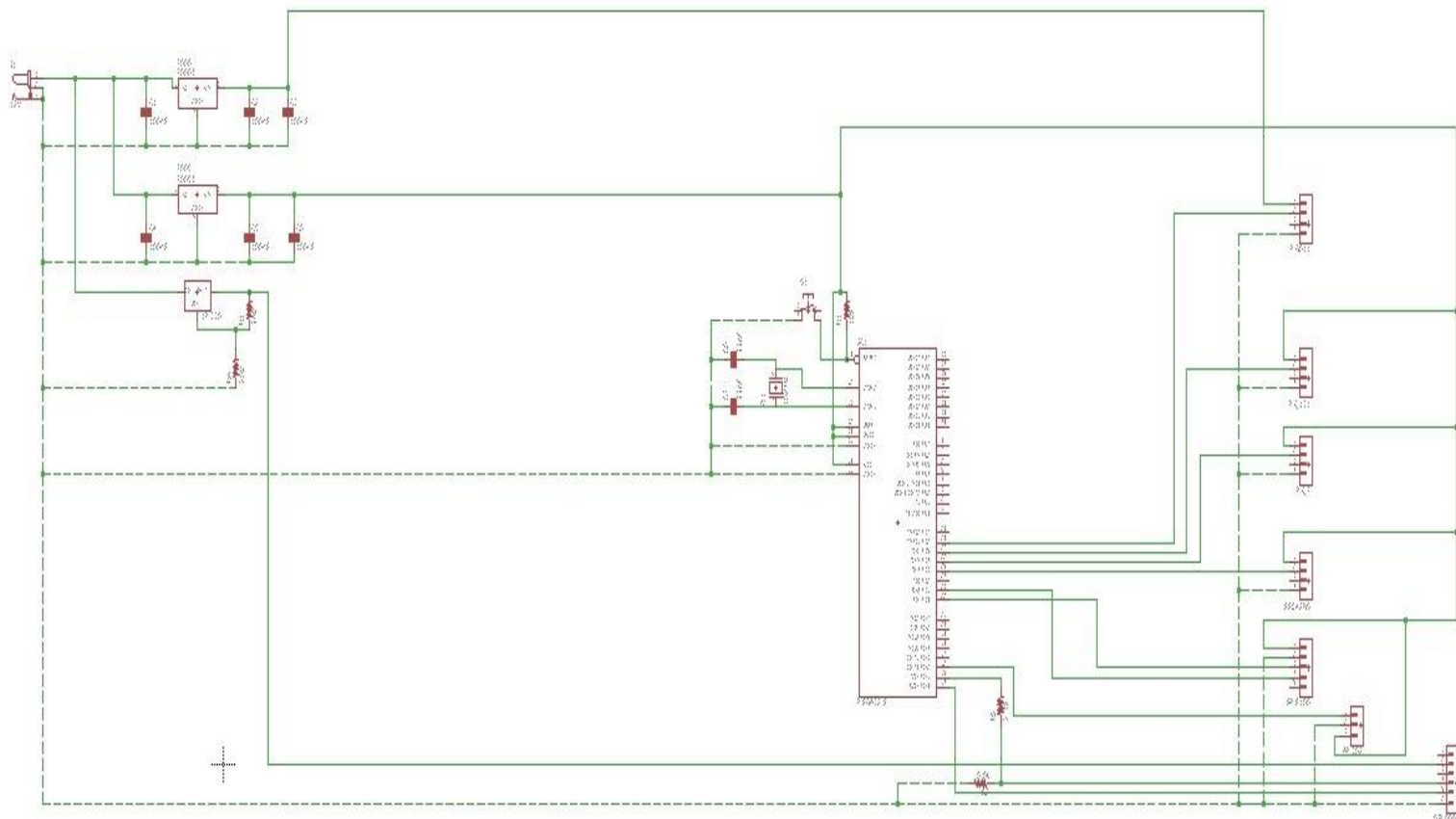
- Evans, D. (11 de 4 de 2011). *Internet of Things*. Obtenido de Cisco: https://www.cisco.com/c/dam/global/es_es/assets/executives/pdf/Internet_of_Things_IoT_IBSG_0411FINAL.pdf.
- Fernández, A. (16 de Enero de 2008). *Efectos nocivos de los compuestos orgánicos volátiles*. Obtenido de http://www.consumer.es: http://www.consumer.es/web/es/medio_ambiente/urbano/2006/11/13/157193.php
- Fundación Tierra. (10 de Marzo de 2017). *Los contaminantes atmosféricos: las partículas en suspensión (PM)*. Obtenido de <http://www.terra.org: http://www.terra.org/categorias/articulos/los-contaminantes-atmosfericos-las-particulas-en-suspension-pm>
- González Aria, A. (23 de Febrero de 2018). *EFFECTOS DEL OZONO SOBRE LA SALUD*. Obtenido de <http://www.vivosano.org: http://www.vivosano.org/efectos-del-ozono-la-salud/>
- Gonzalez, V. (8 de Mayo de 2012). *Diseñan un sensor portátil para medir la calidad del aire*. Obtenido de Nueva Mujer: <https://www.nuevamujer.com/bienestar/2012/05/08/disenan-un-sensor-portatil-para-medir-la-calidad-del-aire.html>
- Gregori, J. (2 de Febrero de 2015). *Un ciclista respira tres veces más contaminación que un peatón*. Obtenido de http://cadenaser.com: http://cadenaser.com/ser/2015/02/17/ciencia/1424190565_502012.html
- Junta de Andalucía. (1996). *Protección Ambiental y sus Reglamentos*. Sevilla: Junta de Andalucía.
- Lindh, W., Pooler, M., Tamparo, C., Dahl, B., & Morris, J. (2013). *Delmar's Comprehensive Medical Assisting: Administrative and Clinical Competencies*. Cengage Learning.
- Loureiro, R. (12 de Junio de 2015). *Estudio Plataformas IoT*. Obtenido de <http://openaccess.uoc.edu: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/42812/6/rloureiroTFC0615 memoria.pdf>
- Mairal, D. (2 de Mayo de 2013). *Consecuencias del CO2 en los humanos*. Obtenido de <http://www.aragonvalley.com: http://www.aragonvalley.com/consecuencias-directas-del-co2-en-los-humanos/>
- Malaver, C. (27 de Junio de 2016). *Investigación recomienda uso de máscara a biciusuarios en Bogotá*. Obtenido de <http://www.eltiempo.com: http://www.eltiempo.com/archivo/documento/CMS-16631425>

- Mario, C. G. (1 de Febrero de 2017). *9,2 % de muertes en el Valle de Aburrá son por contaminación*. Obtenido de <http://www.elcolombiano.com: http://www.elcolombiano.com/antioquia/contaminacion-responsable-de-parte-de-las-muertes-por-causas-naturales-en-medellin-AD5849992>
- Mell, P., & Grance, T. (10 de Julio de 2009). *The NIST Definition of Cloud Computing*. Obtenido de [http://www.nist.gov: nist.gov: http://www.nist.gov/sites/default/files/documents/itl/cloud/cloud-def-v15.pdf](http://www.nist.gov/sites/default/files/documents/itl/cloud/cloud-def-v15.pdf)
- Organización Mundial de la Salud. (2005). *Guías de calidad del aire de la OMS relativas al material particulado, el ozono, el dióxido de nitrógeno y el dióxido de azufre*. Obtenido de http://apps.who.int: http://apps.who.int/iris/bitstream/handle/10665/69478/WHO_SDE_PHE_OEH_06.02_spa.pdf;jsessionid=5DEED72E36365CFC5AB353104BB416F4?sequence=1
- Organización Mundial de la Salud. (2018). *Departamento de Salud Pública, Medio Ambiente y Determinantes Sociales de la Salud*. Obtenido de http://www.who.int: http://www.who.int/phe/health_topics/outdoorair/databases/health_impacts/es/index1.html
- Osborne, A. (18 de Julio de 2017). *Cómo el monóxido de carbono afecta al cuerpo humano*. Obtenido de https://muyfitness.com: https://muyfitness.com/como-el-monoxido-de-carbono-afecta-al-cuerpo-humano_13143407/
- Perujo, E. (28 de 6 de 2016). *Regular la temperatura en el ciclismo*. Obtenido de <https://deportesaludable.com: https://deportesaludable.com/deportes/ciclismo/regular-la-temperatura-ciclismo/>
- Rodríguez, A. M., & Fernández, M. (2009). *La Casa Inteligente*. Obtenido de [it.uc3m.es: it.uc3m.es: http://www.it.uc3m.es/jvillena/irc/practicas/08-09/24.pdf](http://www.it.uc3m.es: it.uc3m.es: http://www.it.uc3m.es/jvillena/irc/practicas/08-09/24.pdf)
- Santoyo, B. (11 de Noviembre de 2014). *Conoce la calidad del aire con AirBeam [Futuro Verde]*. Obtenido de <https://www.nuevamujer.com: https://www.nuevamujer.com/bienestar/2014/11/11/conoce-la-calidad-del-aire-con-airbeam-futuro-verde.html>
- Sanz León, A. (23 de Diciembre de 2015). *Mejorar la calidad del aire interior: Un negocio rentable*. Obtenido de [Prevenblog: http://prevenblog.com/mejorar-la-calidad-del-aire-interior-negocio-rentable/](http://prevenblog.com: http://prevenblog.com/mejorar-la-calidad-del-aire-interior-negocio-rentable/)
- Tanenbaun, A., & Wetherall, D. (2003). *Redes de Computadoras*. Monterrey: PEARSON.
- Treacy, M. (13 de Febrero de 2013). *Air Quality Egg collects and maps real-time air pollution data*. Obtenido de [treehugger.com: https://www.treehugger.com/gadgets/air-quality-egg.html](https://www.treehugger.com: https://www.treehugger.com/gadgets/air-quality-egg.html)

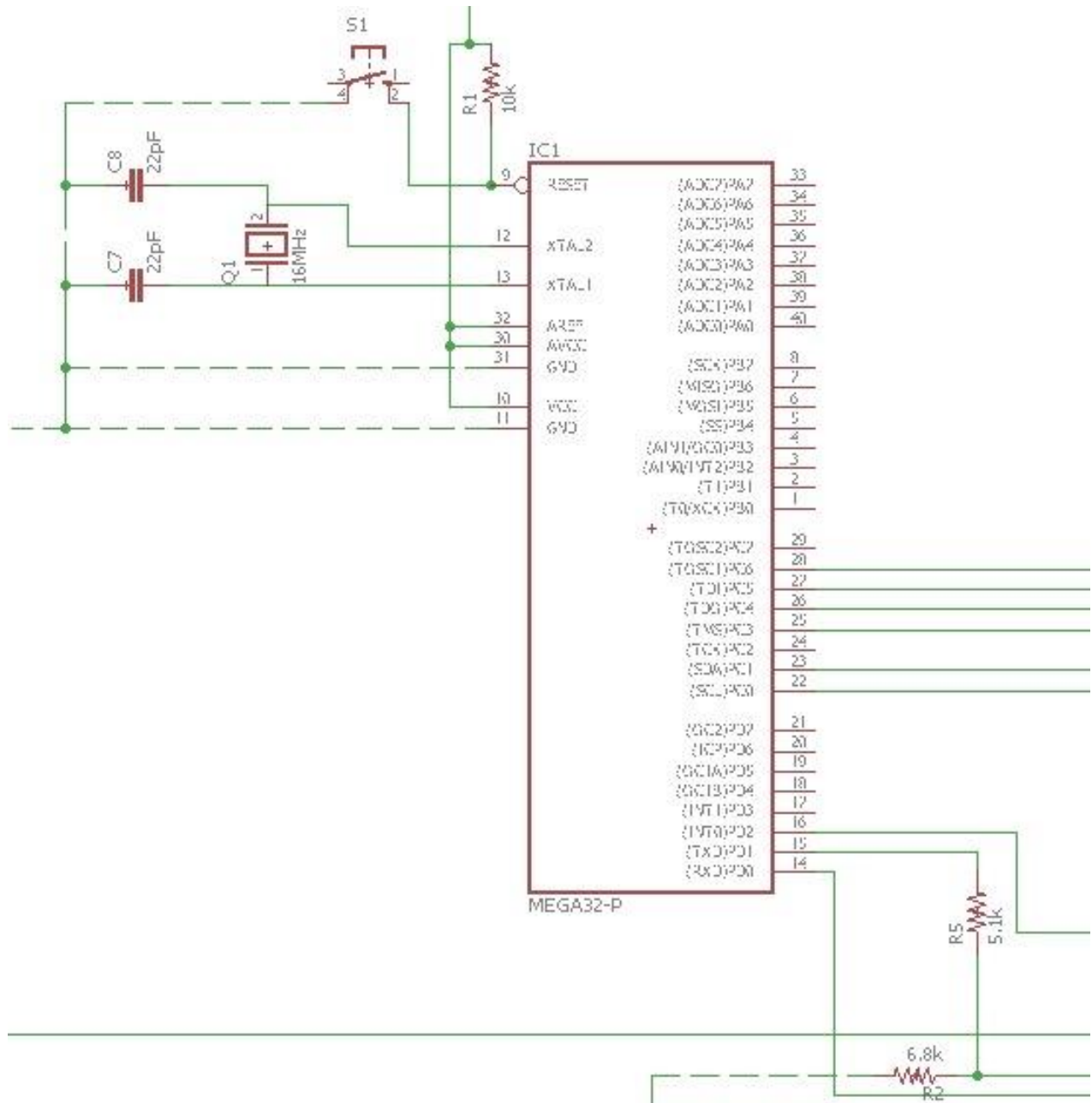
Villa, C. T. (17 de Marzo de 2017). *Metro evitó emisión de 407.817 toneladas de gases contaminantes*. Obtenido de <http://www.elcolombiano.com:>
<http://www.elcolombiano.com/antioquia/metro-de-medellin-evito-emision-de-gases-contaminantes-DC6163225>

Zambrano, D. (6 de Marzo de 2018). *El ingenioso sensor, hecho por paisas, para medir la calidad del aire*. Obtenido de [elcolombiano.com:](http://www.elcolombiano.com:)
<http://www.elcolombiano.com/antioquia/sensor-hecho-por-paisas-mide-la-calidad-del-aire-CC8307499>

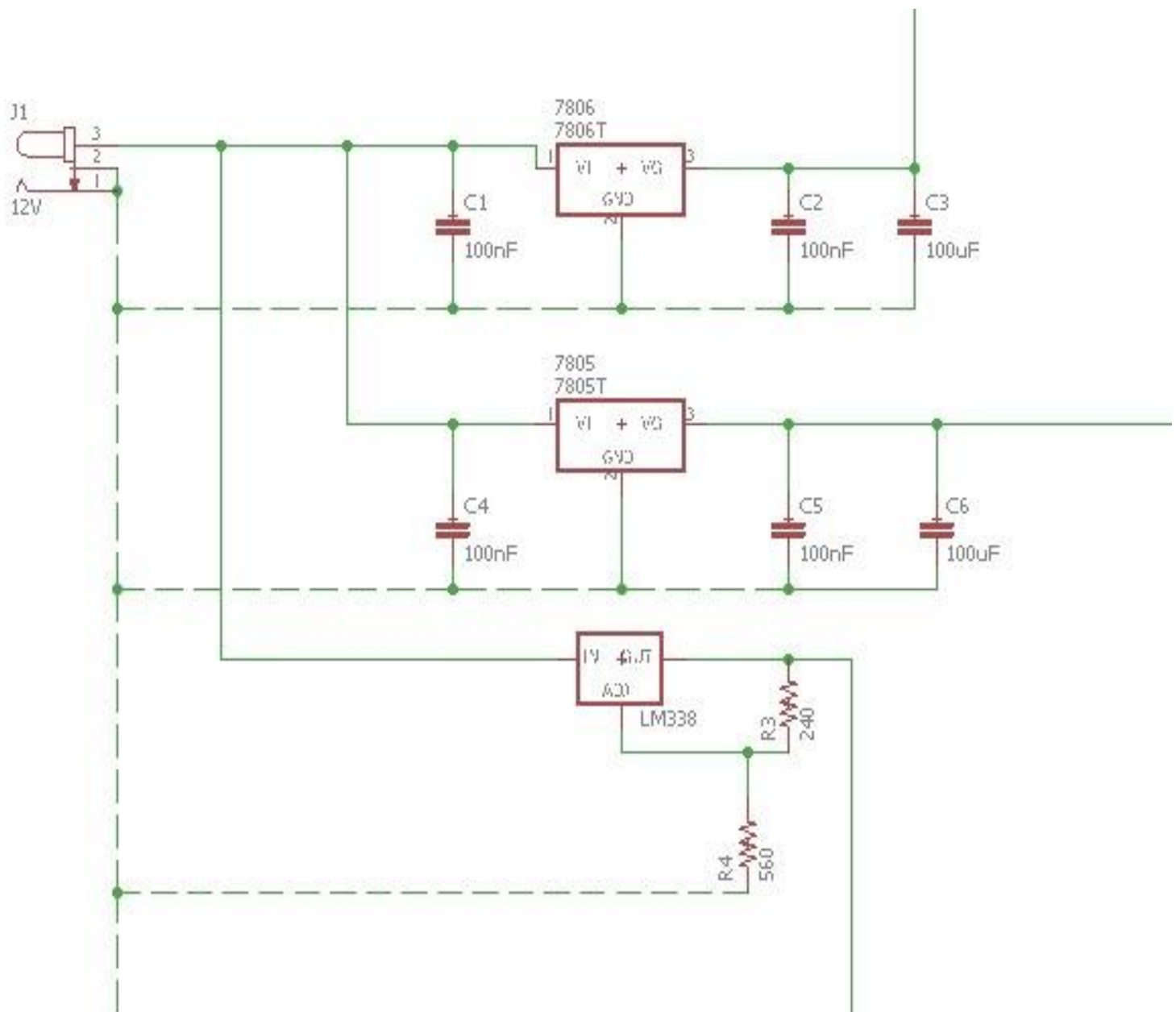
ANEXO 1. DISEÑO ELECTRÓNICO



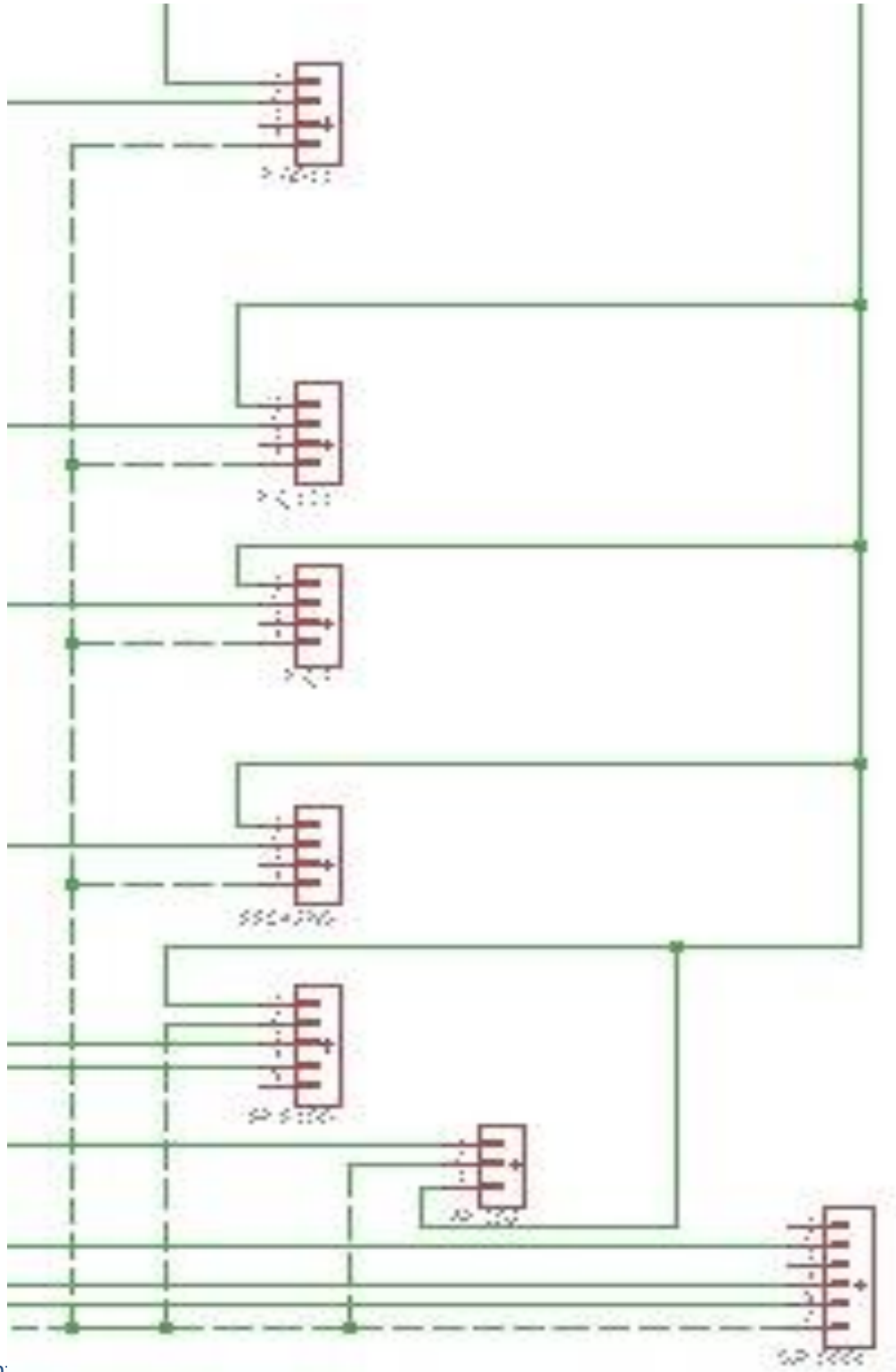
ANEXO 2. DISEÑO ELECTRÓNICO ZOOM 1



ANEXO 3. DISEÑO ELECTRÓNICO ZOOM 2



ANEXO 4. DISEÑO ELECTRÓNICO ZOOM 3



ANEXO 5. DISEÑO ELECTRÓNICO PCB

